

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Радіотехнічний факультет
Кафедра радіотехнічних пристроїв та систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 172 «Телекомунікації та радіотехніка»

Освітньо-професійна програма «Радіосистемна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____Сергій ЖУК

«___»_____20__ р.

ЗАВДАННЯ

на дипломну роботу студенту

Омеляненку Богдану Андрійовичу

1. Тема роботи «Програмне забезпечення для динамічного адаптивного представлення контенту на основі інформаційно-комунікаційної системи «ПОЛІДАР»,

керівник роботи Товкач Ігор Олегович, ст. викладач, затверджені наказом по університету від «21» травня 2020 р. №1126-с

2. Термін подання студентом роботи 15 червня

3. Вихідні дані до роботи веб додаток, на основі мови програмування php та javascript, підтримка десктопної та мобільної версії додатку, зручність інтерфейсу, швидкість в розміщенні контенту, можливість розробки унікального інтерфейсу

4. Зміст пояснювальної записки вступ, аналіз існуючого програмного забезпечення, вибір програмного забезпечення, постановка завдання, висування вимог до програмного забезпечення, розробка програмного забезпечення, охорона праці

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо) три плакати: даталогічна модель, інфологічна модель, зовнішній вигляд сторінки програмного забезпечення

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Гусєв А.М., к.б.н., доцент		

7. Дата видачі завдання 13 квітня 2020р

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Вибір теми обговорення, обговорення можливих варіантів, остаточний вибір теми	13.04.20 – 19.04.20	
2	Пошук існуючих аналогів, виконання порівняльної характеристики аналогів	20.04.20 – 26.04.20	
3	Формування постановки задачі, висування вимог до структури програмного забезпечення	27.04.20 – 03.05.20	
4	Формування власної інфологічної моделі, Формування власної даталогічної структури	04.05.20 – 17.05.20	
5	Розробка зовнішнього вигляду програмного забезпечення	18.05.20 – 07.06.20	

Студент

Богдан ОМЕЛЯНЕНКО

Керівник

Ігор ТОВКАЧ

Анотація

Звіт про дипломну роботу: 83 с., 21 рис., 12 табл., 14 джерел.

Об'єктом розробки і дослідження є процес представлення інформації в доступній та простій формі. Як предмет дослідження розглядається розробка ефективного програмного забезпечення для динамічного адаптивного представлення інформації. Мета роботи – дослідження і розробка структури та програмного забезпечення для динамічного адаптивного представлення контенту.

Проаналізовані сучасні системи створення та управління контентом. Виявлено, що жодна із проаналізованих універсальних CMS систем не відповідають вимогам для зручного динамічного адаптивного представлення інформації, оскільки вони мають ряд суттєвих недоліків, які виправити можливо лише через переписування значних фрагментів програмного коду самих систем. Тому для полегшення розробки за основу було взято інформаційно-комунікаційну систему «ПОЛІДАР». Данна система дає змогу легко запровадити систему авторизації на сайті, підключити систему завантаження файлів та багато інших готових рішень.

Розроблено функціонал ADAPTIVE CMS системи «ПОЛІДАР», який містить в собі усі необхідні інструменти для динамічного адаптивного представлення контенту.

Annotation

Report on diplomatic work: 83 pages, 21 figures, 12 tables, 14 sources.

The object of development and research is the process of presenting information in an accessible and simple form. The subject of the research is the development of effective software for dynamic adaptive presentation of information. The purpose of the work - research and development of structure and software for dynamic adaptive presentation of content.

Modern content creation and management systems are analyzed. It was found that none of the analyzed universal CMS systems do not meet the requirements for convenient dynamic adaptive presentation of information, because they have a number of significant shortcomings that can be corrected only by rewriting large fragments of program code of the systems themselves. Therefore, to facilitate the development, the information and communication system "POLIDAR" was taken as a basis. This system allows you to easily implement an authorization system on the site, connect a file download system and many other ready-made solutions.

The ADAPTIVE CMS functionality of the POLIDAR system has been developed, which contains all the necessary tools for dynamic adaptive presentation of content.

Аннотация

Отчет о дипломной работе: 83 с., 21 рис., 12 табл., 14 источников.

Объектом разработки и исследования является процесс представления информации в доступной и простой форме. В качестве предмета исследования рассматривается разработка эффективного программного обеспечения для динамического адаптивного представления информации. Цель работы - исследование и разработка структуры и программного обеспечения для динамического адаптивного представления контента.

Проанализированы современные системы создания и управления контентом. Выявлено, что ни одна из проанализированных универсальных CMS систем не соответствуют требованиям удобного динамического адаптивного представления информации, поскольку они имеют ряд существенных недостатков, которые исправить можно только через переписывание значительных фрагментов программного кода самих систем. Поэтому для облегчения разработки за основу было взято информационно-коммуникационную систему «ПОЛИДАР». Данная система позволяет легко

внедрить систему авторизации на сайте, подключить систему загрузки файлов и много других готовых решений.

Разработан функционал ADAPTIVE CMS системы «ПОЛИДАР», который содержит в себе все необходимые инструменты для динамического адаптивного представления контента.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломної роботи

на тему: «Програмне забезпечення для динамічного адаптивного
представлення контенту на основі інформаційно-комунікаційної системи
«ПОЛІДАР»

Київ — 2020 року

ЗМІСТ

Перелік символів та скорочень.....	4
Вступ.....	5
1 Аналіз існуючих підходів до вирішення поставленої задачі.....	7
1.1 Поняття системи динамічного адаптивного представлення контенту.....	7
1.2 Функціонал та компоненти CMS.....	11
1.3 Структура сайту, оснований на використанні CMS.....	14
1.4 Існуючі рішення CMS.....	16
1.5 Порівняльний аналіз обраних CMS-систем.....	18
Висновки.....	29
2 «ПОЛІДАР» – паралельна обробка лінійних раціональних даних алгоритмами рекомбінації.....	30
Висновки.....	34
3 Системний підхід до розробки структури програмного забезпечення.....	35
3.1 Постановка задачі.....	36
3.2 Додаткові вимоги до веб-сайту.....	36
3.3 Вимоги до оформлення матеріалів, які викладаються на сайті.....	37
Висновки.....	37
4 Розробка структури програмного забезпечення.....	38
4.1 Опис принципу роботи та функціоналу використаних інструментів.....	38
4.2 Побудова інфологічної моделі предметної області.....	43
4.3 Побудова даталогічної моделі предметної області.....	47
4.4 Розробка сайту програмного забезпечення	54
4.4.1 Сторінка налаштування зовнішнього вигляду.....	55
4.4.2 Головний розділ службової сторінки.....	59
4.4.3 Розділ конфігурації меню службової сторінки.....	60
4.4.4 Розділ налаштування сайту службової сторінки.....	62

4.4.5 Розділ налаштування вмісту пункту меню.....	64
4.5 Файлова структура програмного забезпечення.....	66
4.6 Опис алгоритму створення контенту.....	67
Висновки.....	68
5 Охорона праці та безпека в надзвичайних ситуаціях.....	69
5.1 Визначення основних потенційно небезпечних та шкідливих виробничих факторів.....	69
5.2 Оцінка існуючих умов.....	70
5.3 Комплекс вправ для відновлення.....	73
5.4 Електробезпека.....	77
5.5 Безпека в надзвичайних ситуаціях.....	79
5.6 Пожежна безпека.....	79
ЗАГАЛЬНІ ВИСНОВКИ.....	82
ПЕРЕЛІК ПОСИЛАНЬ.....	83

ПЕРЕЛІК СИМВОЛІВ ТА СКОРОЧЕНЬ

CMS	Content Management System (система керування вмістом)
ПОЛІДАР	Паралельна обробка лінійних ірраціональних даних алгоритмами рекомбінації
PHP	PHP: Hypertext Preprocessor (PHP: гіпертекстовий препроцесор)
MySQL	My Structured Query Language (вільна система керування реляційними базами даних)
HTML	HyperText Markup Language (мова розмітки гіпертекстових документів)
AJAX	Asynchronous JavaScript And XML (динамічне оновлення веб- сторінок)
DOM	Document Object Model (об'єктна модель документа)
SQL	Structured query language (мова структурованих запитів)
MVC	Model–View–Controller (модель–вигляд–контролер)

ВСТУП

Останнім часом інформаційний потік в нашому житті постійно зростає. Відповідно виникає необхідність у зручному зберіганні, упорядкуванні, представленні інформації з можливістю перегляду, зміни, коментування або модифікації.

На сьогоднішній день існує чимала кількість програмного забезпечення, що здатне упорядковувати дані. Дане програмне забезпечення поділяється на дві категорії: офлайн та онлайн. В сучасному світі переважна більшість людей має доступ до інтернету [1]. І тому більший інтерес являє онлайн програмне забезпечення (веб програмного забезпечення), оскільки інформацію можна подати широкому колу людей. Одним із прикладів онлайн програмного забезпечення є веб сайти, які найкращим способом представлення контенту для людей. Тому доцільним є використання веб сайту, як платформи для програмного забезпечення.

На сьогоднішній день існує декілька способів створення веб сайту. Найпростішим є замовлення сайту у спеціалістів, але ціна створення сайту може бути дуже значною. Іншим рішенням проблеми є самостійна розробка сайту. Проте для розробки сайту необхідно володіти певним набором знань у галузі веб програмування, мати навички дизайну та вміння створювати зручний та інтуїтивний графічний інтерфейс. Компромісним рішенням є CMS конструктори, які дозволяють полегшити самостійну розробку веб сайтів. Але без знання веб сервісів (налаштування хостингу, встановлення CMS конструкторів) це все одно буде зробити складно. А також в залежності від різних конструкторів є значна складність при використанні: незручний інтерфейс, великий поріг входження та дуже обмежений функціонал. Також різні конструктори краще підходять для різних типів веб сайтів (форуми, соціальні мережі, інтернет магазини, тощо).

У даній дипломній роботі увагу зосереджено саме на вищеперерахованих недоліках програмного забезпечення для представлення інформації. Отже, метою роботи є дослідження і розробка структури та програмного забезпечення для динамічного адаптивного представлення контенту. Програмне забезпечення має логічну структуру, інтуїтивно зрозумілий інтерфейс та містить в собі всі необхідні інструменти для представлення інформації. Відповідно створення даного програмного забезпечення є доцільним.

1 АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ДО ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

В наш час, час новітніх технологій та значного технічного прогресу найлегшим, найактуальнішим та найдоступнішим джерелом інформації є всесвітня павутина. Інформація в інтернеті подається у вигляді сайтів, вони є універсальним способом представлення практично будь-якої інформації від звичайної текстової до складних мультимедійних систем. На початкових етапах життя всесвітньої павутини, створення сайту зводилася до розробки файлової структури, що складалась з HTML-сторінок із ручним розміщенням тексту, елементів навігації, зображень та посилань. З часом сайти ставали все більшими, складнішими та більш комплексними, це призвело до постійної необхідності виконувати великі об'єми практично однакової рутинної роботи, відповідно виникла потреба в автоматизації. Вирішенням цієї проблеми стали системи управління контентом (CMS), які були розроблені для виконання рутинних, однотипних завдань зі створення та обслуговування веб-ресурсів. Розглянемо найпопулярніші системи керування контентом, їх особливості, переваги та недоліки.

1.1 Поняття системи динамічного адаптивного представлення контенту

Для початку розглянемо поняття та особливості адаптивності.

Адаптивний веб-дизайн (Responsive web design (RWD)) - це підхід до веб-дизайну, завдяки якому веб-сторінки добре відображаються на різних пристроях та розмірах вікон чи екранів. Зміст, дизайн та продуктивність необхідні на всіх пристроях, щоб забезпечити зручність і задоволення.

Сайт, розроблений за допомогою RWD, адаптує макет до середовища перегляду, використовуючи текучі, пропорційні сітки, гнучкі зображення, та медіа-запити CSS3.

Концепція сітки текучого середовища вимагає, щоб розмір елементів сторінки був у відносних одиницях, таких як відсотки, а не в абсолютних одиницях, таких як пікселі чи точки.

Гнучкі зображення також розміщуються у відносних одиницях, щоб не допускати їх відображення поза елемента, що їх містить.

Медіа-запити дозволяють сторінці використовувати різні правила стилю CSS на основі характеристик пристрою, на якому відображається сайт, наприклад ширина поверхні візуалізації (ширина вікна браузера або фізичний розмір дисплея).

Адаптивні макети автоматично налаштовують та адаптуються до будь-якого розміру екрана пристрою, будь то настільний ПК, ноутбук, планшет чи мобільний телефон.

Адаптивний веб-дизайн став більш важливим, оскільки кількість мобільного трафіку зараз складає більше половини загального інтернет-трафіку. Тому Google оголосив Mobilegeddon у 2015 році та почав підвищувати рейтинг сайтів, зручних для мобільних пристроїв, якщо пошук здійснювався з мобільного пристрою. Адаптивний веб-дизайн є прикладом пластичності користувацького інтерфейсу.

Що краще: розробка адаптивної версії сайту або мобільного?

Мобільну версію сайтів доведеться розробляти окремо для кожного виду пристроїв, що істотно позначиться на витратах. Адаптивний дизайн автоматично підлаштовується під будь-яку ширину екрану

Мобільна версія сайтів добре виглядає тільки на смартфонах, нетбуках, електронних книгах, планшетах. Створення адаптивного сайту передбачає можливість зручного перегляду з будь-яких пристроїв, які мають вихід в інтернет.

Мобільна версія сайту виявляється менш результативною з точки зору пошукового просування. Пов'язано це з появою дублів сторінок з піддоменів. У адаптивної версії зберігається один url. Це вигідно для оптимізації в мережі і зручно для користувачів - їм не доведеться запам'ятовувати піддомени.

Мобільна версія здатна істотно скоротити контент або урізати функціонал сайту. Це, як правило, призводить до великих відмов і втрати позицій в пошуковій видачі. Вирішивши створити адаптивний дизайн сайту, ви не втратите ні одного важливого елемента, але інформація на різних пристроях буде показуватися в видозміненому вигляді. Відсоток відмов при цьому значно нижче.

Власники мобільної версії сайту мають низьку конверсію в порівнянні з адаптивною. Оскільки оформляти замовлення на адаптивному сайті зручно з будь-яких пристроїв, відповідно, відсоток продажів товарів або послуг буде вище.

Розглянемо поняття та особливості динамічності веб сторінки.

Динамічна веб-сторінка - це веб-сторінка, побудова якої керується сервером додатків, що обробляє сценарії на стороні сервера. У сценарії на стороні сервера параметри визначають, як проходить збірка кожної нової веб-сторінки, включаючи налаштування більшої кількості обробних даних на стороні клієнта.

Динамічна веб-сторінка на стороні клієнта обробляє веб-сторінку, використовуючи сценарій HTML, що працює в браузері під час завантаження. JavaScript та інші мови сценаріїв визначають спосіб аналізу HTML на отриманій сторінці та модель об'єкта документа або DOM, що представляє завантажену веб-сторінку. Потім ті самі методи клієнтської динаміки можуть динамічно оновлювати або змінювати DOM таким же чином. Незважаючи на те, що веб-сторінка може бути динамічною на стороні клієнта, вона все ще може розміщуватися на статичному сервісі хостингу,

такому як GitHub Pages або Amazon S3, доки не викликаний код, що знаходиться на стороні сервера.

Потім динамічна веб-сторінка завантажується користувачем або комп'ютерною програмою для зміни деякого змінного змісту. Інформація про оновлення може надходити з сервера або із змін, внесених у DOM цієї сторінки. Використовуючи технології Аїах, кінцевий користувач отримує одну динамічну сторінку, керовану як одна сторінка у веб-браузері, а фактичний веб-вміст, що відображається на цій сторінці, може змінюватися. Двигун Аїах працює лише в браузері, вимагаючи частини свого DOM, DOM, для свого клієнта, із сервера додатків.

Класична навігація по гіпертексту, лише в HTML або XHTML, надає "статичний" вміст, тобто користувач запитує веб-сторінку та просто переглядає сторінку та інформацію на цій сторінці.

Однак веб-сторінка може також забезпечити користувальницьку функцію "живого", "динамічного" або "інтерактивного". Вміст (текст, зображення, поля форми тощо) на веб-сторінці може змінюватися у відповідь на різні контексти чи умови.

Існує два способи створення такого ефекту:

Використання сценаріїв на стороні клієнта для зміни поведінки інтерфейсу на певній веб-сторінці у відповідь на дії миші або клавіатури або на визначені події часу. У цьому випадку динамічна поведінка відбувається в рамках презентації.

Використання сценаріїв на стороні сервера, щоб змінити надане джерело сторінки між сторінками, регулюючи послідовність або перезавантажувати веб-сторінки або веб-контент, що постачаються до браузера. Відповіді сервера можуть визначатися такими умовами, як дані у розміщеній HTML формі, параметри в URL-адресі, тип використовуваного браузера, проходження часу або стан бази даних або сервера.

Веб-сторінки, які використовують сценарій на стороні клієнта, повинні використовувати технологію презентації. Клієнтські мови сценаріїв, такі як JavaScript або ActionScript, які використовуються відповідно для технологій Dynamic HTML (DHTML) та Flash, часто використовуються для упорядкування типів медіа (звук, анімація, зміна тексту тощо). Сценарій також дозволяє використовувати віддалений скрипт, метод, за допомогою якого сторінка DHTML вимагає додаткової інформації від сервера, використовуючи прихований кадр, XMLHttpRequests або веб-сервіс.

Веб-сторінки, які використовують сценарії на стороні сервера, часто створюються за допомогою серверних мов, таких як PHP, Perl, ASP, ASP.NET, JSP, ColdFusion та інших мов. Ці серверні мови зазвичай використовують загальний інтерфейс шлюзу (CGI) для створення динамічних веб-сторінок. Ці типи сторінок можуть також використовувати на стороні клієнта першого типу (DHTML тощо).

1.2 Функціонал та компоненти CMS

Для того, щоб краще розглянути окремі CMS і відмінності між ними, слід надати визначення CMS і її функціоналу.

Необхідність систем управління для власників сайтів почала даватися взнаки в той момент, коли кількість матеріалів на веб-сайтах почала стрімко зростати. Це призвело до того, що традиційні «ручні» технології розробки і підтримки сайтів, коли сайт складався з статичних сторінок і набору додаткових спеціалізованих скриптів, стали не встигати за швидко змінними умовами бізнесу. Введення даних на сайт вимагало (як мінімум) знання технологій HTML / CSS верстки, зміни структури сайтів були пов'язані з каскадним зміною великої кількості взаємозв'язаних сторінок. Різні автоматизовані механізми, на зразок гостьових книг і новинних стрічок, упроваджені на сайтах як окремі скрипти і, як правило, написані різними

фахівцями, перестали задовольняти вимогам безпеки. На багатьох сайтах стали з'являтися проблеми, що пов'язані з наявністю великої кількості різних технологій і підходів до розробки, тому виникла потреба з одного боку в уніфікації програмних рішень, а з іншого в розділенні дизайну і вмісту на дві незалежні складові.

Абревіатура CMS розшифровується як «Content Management Software» («програмне забезпечення для управління вмістом»). Іноді вживається більш просту назву - "движок сайту". CMS з'явилися не так давно. Першою системою прийнято вважати Vignette, яка з'явилася на заході в 1995 році.

Отже, система управління контентом (CMS) - це програмне забезпечення, яке можна використовувати для управління створенням і модифікацією цифрового контенту. CMS зазвичай використовуються для управління контентом підприємства (ECM) та управління веб-контентом (WCM). ECM, як правило, підтримує декількох користувачів у середовищі спільної роботи, інтегруючи управління документами, управління цифровими активами та збереження записів. Крім того, WCM є спільним автором для веб-сайтів і може включати текст та вбудовувати графіку, фотографії, відео, аудіо, карти та програмний код, які відображають вміст та взаємодіють із користувачем. ECM зазвичай включає функцію WCM.

Як правило, CMS містить два основні компоненти: додаток для управління вмістом (CMA), як інтерфейс користувача, що дозволяє користувачеві, навіть маючи обмежений досвід, додавати, змінювати та видаляти вміст із веб-сайту без втручання веб-майстра та додаток для доставки вмісту (CDA), який збирає вміст та оновлює веб-сайт.

Основними функціями CMS є: індексація, пошук та виправлення, управління форматом, ревізійний контроль та загальне управління.

Особливості можуть відрізнятися залежно від системного додатку, але зазвичай включають:

Інтуїтивно зрозумілі функції індексації, пошуку та виправлення, індексують усі дані для легкого доступу через функції пошуку та дозволяють користувачам здійснювати пошук за такими атрибутами, як дати публікації, ключові слова чи автор.

Функції редагування дозволяють оновлювати та редагувати вміст після початкової публікації. Контроль редагування також відстежує будь-які зміни, внесені до файлів окремими особами.

Функціонал публікації дозволяє користувачам використовувати шаблон або набір шаблонів, затверджених організацією, а також інші інструменти для створення або зміни вмісту.

Популярні додаткові функції можуть включати:

SEO-дружні URL-адреси. Комплексна та онлайн-допомога, включаючи дошки для обговорень. Групові системи дозволів. Повна підтримка шаблонів та налаштування шаблонів. Панель адміністратора з підтримкою декількох мов. Ієрархія вмісту з необмеженою глибиною та розміром. Мінімальні вимоги до сервера. Інтегровані файлові менеджери. Інтегровані журнали аудиту.

Таким чином, системи управління контентом вирішують дві основні задачі. З точки зору користувачів - це інструмент, який дозволяє публікувати новини, розміщувати нові сторінки на сайті і проводити інші операції над вмістом через зручний інтерфейс. При цьому користувач може не володіти технологіями інтернет-розробки, але він обов'язково повинен розуміти, як влаштований сайт.

З точки зору тих, хто робить сайти - це інструмент, який прискорює розробку складних сайтів, дозволяючи компонувати рішення з готових блоків, змінюючи в певних межах логіку роботи і оформлення.

Як правило, система управління це окремий інтерфейс, призначений для управління сайтом. Він може бути реалізований як веб-додаток, коли користувач входить в адмінзону свого сайту через звичайний браузер за

певною адресою, або як окреме Windows-додаток, що вимагає інсталяції. Різні підходи мають різні переваги і недоліки.

1.3 Структура сайту, основанийо на використанні CMS.

В основі даної технології лежить трирівнева архітектура клієнт / сервер. Така архітектура розбиває процес обробки даних між:

- клієнтом;
- сервером додатків;
- сховищем даних.

На відміну від традиційної дворівневої архітектури тут присутній сервер додатків як проміжна ланка між клієнтом і сховищем даних.

Необхідність такого ланки продиктована вимогами додатків для підприємств, що взаємодіють з клієнтами (через Інтернет), партнерами (через Extranet) і власними працівниками (через Intranet). У відсутності сервера додатків більшість додатків виконується прямо на клієнтському комп'ютері, з якого клієнт посилає запити. При цьому для доступу до необхідних даних клієнт повинен знати, як саме вони організовані і де зберігаються. До того ж комп'ютери клієнтів повинні бути досить потужними для обробки даних зі сховища (така архітектура зазвичай називається "товстий клієнт"). На противагу їй трирівнева архітектура, її ще називають "товстий сервер", а частіше - "тонкий клієнт", дозволяє уникнути посилення комп'ютерів клієнтів і завантаженості мережі через переміщення даних. Отримуючи запит, сервер додатків обробляє його, зв'язуючись зі сховищем даних, в якому б місці необхідні дані не знаходилися. Клієнт лише отримує результат у вигляді HTML-файлу. Таким чином, сервер додатків є стандартизованої платформою для динамічної доставки контенту і побудови основних додатків. Треба

зауважити, що самих серверів додатків може бути багато, а зв'язок з ними відбувається через Web-сервер.

Як було сказано вище, для створення і обслуговування динамічних сайтів можна скористатися спеціальними системами управління Web-контентом. Вони є програмним забезпеченням, що встановлюються на Web-сервері. Їх основним завданням є контроль контенту, що надходить на сайт, для забезпечення достовірності та своєчасності.

Якщо подивитися тепер на системи управління контентом в найзагальнішому вигляді, то можна відзначити, що їх складовими можуть бути компоненти управління контентом, сховища даних, серверів додатків. З них тільки управління контентом є обов'язковою складовою, хоча різні виробники і тут пропонують різні можливості.

Компоненти CMS:

Розробка контенту

Розробка контенту є одним з ключових компонентів всієї системи. Саме тут починається життєвий цикл будь-якого матеріалу публікується на сайті. На цьому етапі відбувається створення, редагування та затвердження контенту, а роль системи полягає в автоматизації цих процесів. Завдання підтримки спільної роботи авторів, редакторів, програмістів і менеджерів повністю перекладається на систему. Це завдання здійснюється завдяки поділу контенту і дизайну. Всі компоненти сайту, включаючи шаблони і наповнення, зберігаються в певних місцях сховища даних. Система ж автоматично звертається в потрібні місця сховища, дозволяючи безлічі користувачів, навіть не є технічними фахівцями, працювати над підготовкою контенту до публікації, включаючи перевірку його достовірності.

Управління сайтом

Інвестиції в систему управління Web-контентом могли вберегти британський каталог Argos від жахливої помилки. На своєму сайті по електронній комерції був представлений телевізор Sony всього за £ 3 (\$ 5).

Помилка сталася, коли один зі співробітників округлив ціну з 299 до 300 фунтів і випадково стер нулі. Система управління контентом могла б перевірити цю інформацію до опублікування.

На цьому рівні відбувається розробка самого сайту, попередній перегляд і публікація підготовленого контенту. Тут розробляється зовнішній вигляд, готуються шаблони, розподіляються ролі користувачів і класифікація необхідної бізнес-інформації (наприклад, товари, ціни). Важливими компонентами цього рівня є служби підтримують своєчасність надходження необхідного контенту.

Доставка контенту

Коли сайт повністю підготовлений до публікації, необхідні кошти для динамічного формування Web-сторінок в залежності від виду конкретних користувачів. У зв'язку з цим, одним з важливих компонентом даного етапу є персоналізація або розподіл профілів, щоб кожен користувач отримував тільки ту інформацію, яка відповідає його ролі.

1.4 Існуючі рішення CMS

На сьогодні існує дуже багато систем управління контентом. Проте близько 90% від них займають менше 1% від усього ринку. Станом на 17 квітня 2020 року 43.4% від усіх веб сайтів створені без використання будь-якої сторонньої системи управління контентом. Тобто більше половини існуючих сайтів тим чи іншим чином були створені на основі систем з управління контентом.

Нижче наведений графік [2] найрозповсюдженіших систем управління контентом:

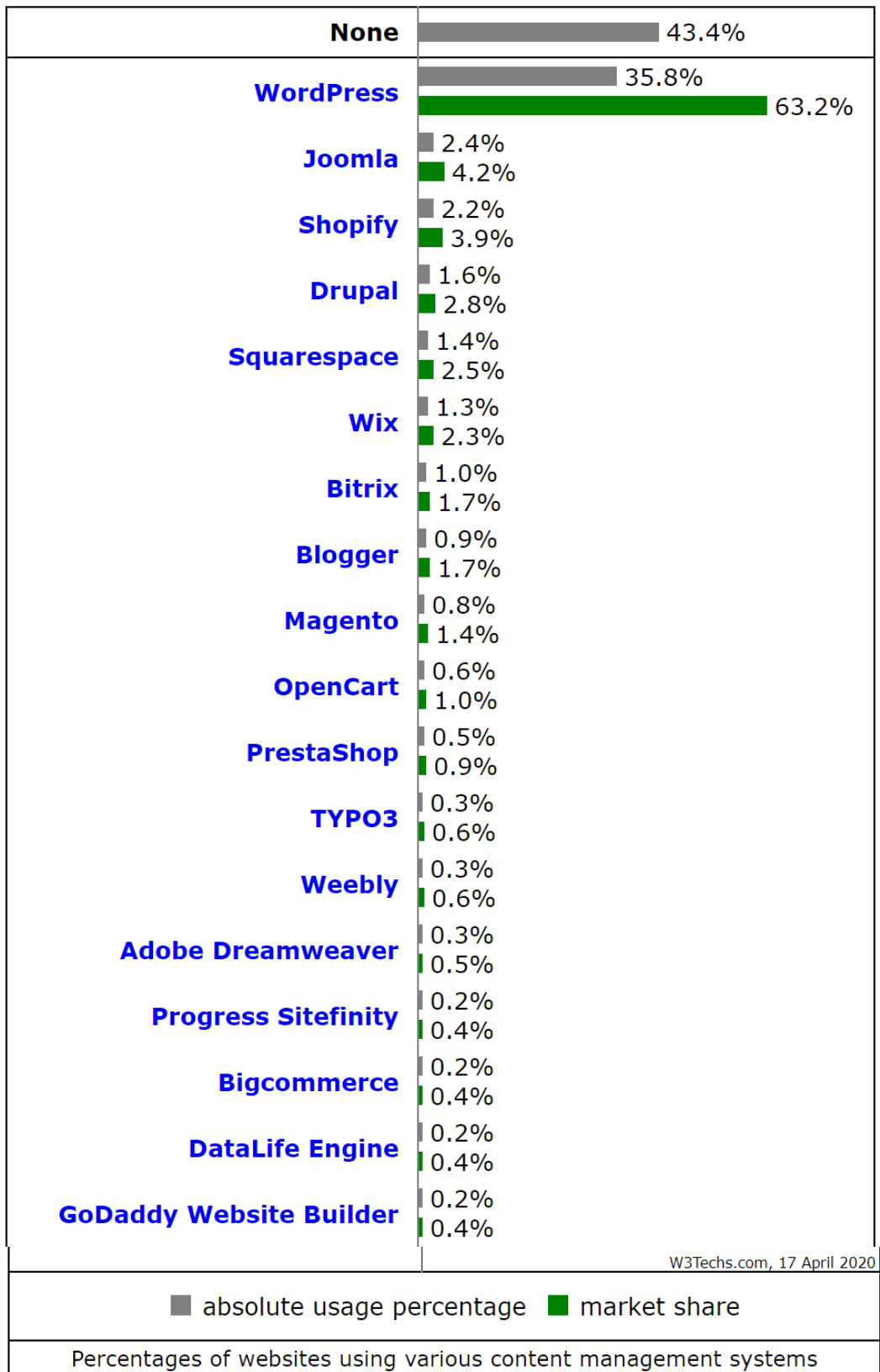


Рисунок 1.4 – Статистика використання CMS систем

Данні наведені станом на 17 квітня 2020 року. На графіку не вказано більше 400 існуючих систем через те, що їх доля на ринку є надзвичайно малою (менше 0.1%).

Як бачимо на ринку існує безліч різноманітних CMS систем. Проблема полягає в тому, що переважна більшість систем є дуже нішевими, тобто вони були створені для виконання дуже вузько направлених завдань, або для розробки лише якоїсь певної частини сайту (системи авторизації на сайті, системи представлення фотографій та ін.). Також багато з цих систем перестали регулярно оновлюватися, тому їх актуальність практично рівна нулю, проте свого часу деякі сайти були створені на їх основі і до сих пір функціонують та представляють певний функціонал та інформацію.

Відповідно такі системи не підходять для порівняння з нашим продуктом. Тому з усього різноманіття систем було обрано лише декілька, що найближче підходять для порівняння з нашим продуктом. Критерієм їх вибору стали такі властивості: можливість створення сайту цілком (а не якоїсь його частини), відносна зручність та доступність у використанні, підтримка останніх веб технологій та браузерів, значна доля на ринку подібних систем.

За даними критеріями було обрано наступні системи: Wix, Бітрікс24, uKit, Google Сайти, Vigbo, Tilda.

Ці системи будуть приймати участь у подальшому порівняльному аналізі з нашою системою.

1.5 Порівняльний аналіз обраних CMS-систем

Як було зазначено в минулому розділі порівняльний аналіз буде проводитися серед наступних систем: Wix, Бітрікс24, uKit, Google Сайти, Vigbo та Tilda. Для початку розглянемо опис кожної з систем, їх ціни,

інтеграції з іншими сервісами, базові можливості та функціонал, платформи на яких можливе створення продукту, доступні мови.

Кожний з розглянутих продуктів буде порівнюватися за наявністю та функціоналом таких можливостей:

Конструктори сайтів: Дисковий простір (Гб), Резервне копіювання, HTML-редактор, SEO-інструменти, Інтернет-магазин, Email-розсилка, Форми зворотного зв'язку, Опитування, Пошук по сайту, Системи оплати для сайтів, Вбудована CRM, Слайдер, Таймери, Шаблони, Редактор зображень, Мультилендінг, А/Б тестування, База знань, Особистий кабінет користувача.

Landing Page: Візуальний редактор, Шаблони, Кількість створюваних сторінок, Функції CRM, Прив'язка домену, Статистика, А/Б тестування, Робота з соціальними мережами, Прийом платежів.

Безпека та конфіденційність: Доступ за протоколом HTTPS, Резервне копіювання.

Загалом наведені продукти мають більшість із перерахованих функцій, тому будемо лише вказувати відсутні функції та наводити точну інформацію по окремим з них (наприклад об'єм даних дискового простору).

Система Wix

Опис: Wix - це безкоштовний онлайн-сервіс з великою кількістю додаткових платних можливостей, який дозволить створити професійний сайт або інтернет-магазин. Розробники Wix пропонують також підключення мобільної версії сайту, редактор, багато шаблонів HTML5, техпідтримку і навіть інструменти пошукової оптимізації [3].

Можливості які пропонує Wix пропонує своїм клієнтам:

Десятки цікавих шаблонів. Оптимізацію сайту під мобільні пристрої. Зрозумілий редактор елементів сайту. Готові палітри, а також порожні і односторінкові шаблони. Інструменти для редагування текстів на сайті. Опції для різноманітного редагування зображень. Внесення додаткового коду. Ефекти переходів. Обмеження доступу до певних сторінок. Надійність

способів оплати в інтернет-магазині. Зручне редагування цін, товарів, широка настройка і управління доставкою і податками. Підтримка тегів. Додавання блогу і соціальних сервісів. Електронні розсилки. Календар та форми. Підтримка Open Graph. Генерація карти сайту. Інструменти для управління рекламою. Корпоративна пошта. Галерея. І багато іншого.

Ціни:

Є наявність пробного періоду, безкоштовного тарифу. Спосіб оплати: за підпискою. Вартість від 48 грн на місяць. Є 14 днів пробного терміну.

Інтеграції з іншими сервісами:

Данна система підтримує наступні сервіси. ЯндексГроші, Google Analytics, LiveChat, Wufoo, Instagram.

Можливості:

Дисковий простір (Гб) – 20. Відсутні: Пошук по сайту, Вбудована CRM, Слайдер, Таймери.

Платформи:

Android, Windows Phone, IOS, веб-сайт.

Доступні мови:

Російська, Англійська.

Система Бітрікс24

Опис:

Бітрікс24 - повний комплект інструментів для роботи компанії. Включає CRM для продажів, завдання та проекти, контакт-центр, конструктор сайтів та інтернет-магазинів, соціальна мережа компанії. Безкоштовний тариф для невеликих компаній до 12 співробітників. CRM допомагає продавати більше: бере під контроль всі канали комунікацій з клієнтами, підказує, що робити і автоматизує продаж. Задачі і проекти допомагають працювати разом і встигати вчасно. Якщо завдання поставлене в Бітрікс24, вона буде виконана. Контакт-центр об'єднує в CRM канали комунікацій з клієнтами: телефон, email, чат на сайті, Facebook, Instagram, ВКонтакте, Яндекс.Чат і інші.

Конструктор сайтів та інтернет-магазинів - сайт повинен бути не просто красивим, який легко створити самому і безкоштовно. Головне завдання сайту - приводити клієнтів. Офіс допомагає працювати разом: чат, диск, календар, групи, робочі звіти, бізнес-процеси і інші інструменти для спільної роботи [4].

Ціни:

Сервіс безкоштовний для компаній до 12 співробітників. При більшій кількості співробітників чи при бажанні користуватися додатковим функціоналом потрібно переходити на платний тариф. Спосіб оплати: за підпискою. Вартість від 387 грн на місяць.

Інтеграції з іншими сервісами:

Данна система підтримує наступні сервіси. ПланФакт, Яндекс.Диск, Яндекс.Почта, Kickidler, WhatsApp Messenger, Calltouch, Telegram Messenger, МойСклад, ПланФикс.

Можливості:

Дисковий простір (Гб) – необмежено. Присутні всі з наведених можливостей.

Платформи:

Android, Windows Phone, IOS, MacOS, веб-сайт.

Доступні мови:

Російська, Англійська, Українська.

Система uKit

Опис:

uKit - це сервіс для створення комерційних сайтів: візиток, магазинів (Eswid Shop) і лендінгів. Оформлення сторінок відбувається в якісному і зручному візуальному редакторі за допомогою перетягування на них віджетів (функціональних елементів). Останні мають власні настройки, які різняться в залежності від їх типу. Приклади віджетів: форма, плеєр, соціальна панель, картка товару, прайс, слайдер, галерея і так далі [5].

Система має вбудовані засоби аналітики - статистику, SEO, Google Analytics, Яндекс.Метрика. Також конструктор надає поради з доопрацювання проблемних областей сайту, що є дуже корисним для початківців. Можливості підключення свого домену, фавікона і логотипу присутні.

Особливості uKit: Простота WYSIWYG-редактора з підтримкою drag'n'drop, Швидка індексація сайтів пошуковими системами, Красиві шаблони, розбиті на категорії, Зручний сайдбар «Add This», сумісний з переважною більшістю соціальних сервісів, Функція замовлення зворотного дзвінка і інтеграції розташування компанії з Яндекс.Карт, Велика кількість віджетів, логічно відсортованих за категоріями, Налаштування SEO та інших параметрів просування, Можливість замовлення у розробників готового сайту за 50 \$.

uKit відмінно підходить для самостійного створення бізнес-візиток і лінденгів. Він має якісну технічну підтримку і велику кількість підказок всередині системи. Сайтбілдер дозволяє досягати якісного результату мінімальними зусиллями.

Ціни:

Є наявність пробного періоду, безкоштовного тарифу. Спосіб оплати: за підпискою. Вартість від 133 грн на місяць. Є 14 днів пробного терміну.

Інтеграції з іншими сервісами:

Данна система підтримує наступні сервіси. Google Analytics, Wufoo, Instagram.

Можливості:

Дисковий простір (Гб) – необмежено. Відсутні: Email-розсилка, Опитування, Системи оплати для сайтів, Вбудована CRM, Особистий кабінет користувача.

Платформи:

Лише веб-сайт.

Доступні мови:

Російська, Англійська.

Система Google Сайти

Опис:

Google Сайти - це невеликий веб-сервіс для створення сайтів без необхідності програмувати. Візуальний редактор дозволяє створити сайт так само просто, як текстовий документ - можна обійтися без допомоги фахівців або розробників [6].

Сервіс підходить для малого бізнесу, ведення блогів, корпоративних сайтів, приватних сайтів під заходи. Доступна SEO-оптимізація від фахівців Google. Реалізована підтримка віджетів, блоків з календарем, відео, карт, презентацією та будь-яких форматів сервісів Google.

При створенні сайту використовуються пошукові технології Google, є управління доступом для командної роботи.

Ціни:

Немає пробного періоду. Є безкоштовний тариф. Спосіб оплати: за підпискою. Безкоштовний, проте можна замовити підтримку додаткових функцій від 144 грн на місяць.

Інтеграції з іншими сервісами:

Данна система підтримує наступні сервіси. Всі сервіси від Google.

Можливості:

Дисковий простір (Гб) – 15. Відсутні: Резервне копіювання, SEO-інструменти, Інтернет-магазин, Email-розсилка, Опитування, Системи оплати для сайтів, Вбудована CRM, Таймери, Шаблони, Редактор зображень, Мультилендінг, А/Б тестування, База знань, Особистий кабінет користувача.

Платформи:

Android, Windows Phone, IOS, веб-сайт.

Доступні мови:

Українська, Російська, Англійська.

Система Vigbo

Опис:

Vigbo (раніше Gophotoweb) - це сервіс для створення сайту, інтернет-магазину або блогу. За допомогою сервісу користувач може швидко купити готовий сайт і налаштувати його під проект. У візуальному редакторі можна вибирати і перетягувати на сторінку заголовки, тексти, фото і відео, кнопки, стрічку інстаграму, карти, декоративні роздільники та інші блоки. Відмінність від інших конструкторів - готові шаблони для різних типів сторінок: Про мене, Новини, Події, Ціни, Відгуки, Про команду і інші. Кожен шаблон можна відредагувати і налаштовувати під сайт [7].

Основна аудиторія - творчі проекти і малі бізнеси. Серед них - фотографи, дизайнери, архітектори, ювеліри, hand-made проекти.

Можливості Vigbo: Інтернет-магазин з товарами, Можливість приймати оплату через сайт, Каталог дизайнів, Заміна дизайну «в один клік», Готові шаблони для різних типів сторінок, Блог, SEO-настройки, Хостинг сайту, Необмежений дисковий простір, Пошта з прив'язкою до домену, Технічна підтримка, Налаштування логотипу, меню і підміню, Фото- та відео-галереї, Посилання на соціальні мережі.

Ціни:

Є наявність пробного періоду. Немає безкоштовного тарифу. Спосіб оплати: за підпискою. Вартість від 160 грн на місяць.

Інтеграції з іншими сервісами:

Данна система підтримує наступні сервіси. PayOnline та MailChimp.

Можливості:

Дисковий простір (Гб) – необмежено. Відсутні: Особистий кабінет користувача, Резервне копіювання, Опитування, Landing Page, Доступ за протоколом HTTPS, Резервне копіювання в декількох місцях.

Платформи:

Лише веб-сайт.

Доступні мови:

Лише російська.

Система Tilda

Опис:

Tilda Publishing - це веб-платформа, яка дозволяє створювати сайти, інтернет-магазини, лендінгт і блоги без програмування. Головна особливість сервісу - блоковий редактор. Сторінки на Тільда можна збирати з блоків, в бібліотеці вже більше 500 готових варіантів. Вони розділені на 28 категорій: обкладинка, послуги, форми, відгуки, магазин, про компанії [8].

Всі блоки спроектовані дизайнерами і мають правильні відступи і співвідношення розмірів. Кожен блок адаптивний і автоматично підлаштовується під екрани різних пристроїв. Текст і зображення можна змінювати прямо на сторінці в режимі редагування. Зовнішній вигляд можна повністю змінити - для цього є всі настройки.

Для професійних дизайнерів є редактор Zero Block. Він призначений для створення нових унікальних блоків. Інтерфейс буде звичний користувачам популярних графічних редакторів. В Zero Block можна створювати складну анімацію по кроках. Це дозволяє робити інтерактивні сайти без програмування.

Для створення інтернет-магазинів є окрема категорія блоків з кошиком і картками товарів. В кошик можна додати промокод і варіанти доставки з різною ціною. Є інтеграція з 13 платіжними системами: Яндекс Каса, CloudPayments, Robokassa, Яндекс Гроші, PayPal, Stripe, еквайринг Альфа-Банку, Ощадбанку і Тінькофф банку, bePaid BY, LiqPay, 2Checkout.

Заявки з форм можна передавати в месенджери Telegram і Slack, на e-mail, в Google Таблицю, Trello, CRM-системи і сервіси розсилок. Всього можна підключити 19 сервісів. В налаштуваннях сайту в кілька кліків можна підключити лічильник Яндекс Метрики, Google Analytics і Google Tag Manager.

Для SEO можна налаштувати title і description сторінки, додати теги h1, h2, h3 до заголовків і атрибут alt до зображень. Сторінкам можна надати зрозумілі адреси. Окремі сторінки можна закрити від індексації. Файли robots.txt і sitemap.xml генеруються автоматично. Для перевірки помилок в налаштуваннях сайту є окремий інструмент Tilda-панель веб-майстра.

Ціни:

Є наявність пробного періоду, безкоштовного тарифу. Спосіб оплати: за підпискою. Вартість від 195 грн на місяць.

Інтеграції з іншими сервісами:

Данна система підтримує наступні сервіси. Яндекс. ПочтаЯндекс. Telegram Messenger. Битрикс24. SendPulse. RoistatRoistat. SlackSlack JivoSite. Ecwid. Trello. Google Таблицы. GetResponse. 1С. Яндекс.Деньги. retailCRM. Robokassa. Pipedrive. OneBox. Та інші.

Можливості:

Дисковий простір (Гб) – необмежено, Кількість створюваних сторінок – 500. Відсутні: Особистий кабінет користувача

Платформи:

Лише веб-сайт.

Доступні мови:

Англійська, Російська

На основі аналізу обраних CMS-систем складемо таблицю з основними перевагами та недоліками цих систем.

Таблиця 3.1 – Порівняльна характеристика

Система Wix		
Переваги	Недоліки	Ціна
- Підтримка значної кількості можливостей та функцій.	- Розмір дискового простору у 20 Гб. - Відсутність підтримки	від 48 грн на місяць. Є 14 днів пробного терміну.

<ul style="list-style-type: none"> - Наявність пробного періоду. - Доступні інтеграції зі сторонніми сервісами. - Можливість розробки сайту на інших платформах (Android, IOS) 	<ul style="list-style-type: none"> української мови. - Незначна кількість підтримуваних сторонніх сервісів. 	
Система Бітрікс24		
Переваги	Недоліки	Ціна
<ul style="list-style-type: none"> - Підтримка всіх можливостей та функцій (що розглядалися) - Можливість розробки сайту на інших платформах (Android, IOS, MacOS) - Необмежений розмір дискового простору (за доплату) 	<ul style="list-style-type: none"> - Підтримувані сторонні сервіси є загалом російськими і більшість з них не працює в Україні (Яндекс.Диск, Яндекс.Почта, МойСклад, ПланФикс) 	Сервіс безкоштовний для компаній до 12 співробітників.
Система uKit		
Переваги	Недоліки	Ціна
<ul style="list-style-type: none"> - Необмежений розмір дискового простору (за доплату) - Загалом зручний інтерфейс створення сторінок. 	<ul style="list-style-type: none"> - Мала кількість підтримуваних сторонніх сервісів. - Відсутність підтримки української мови. 	Вартість від 133 грн на місяць. Є 14 днів пробного терміну.

	<ul style="list-style-type: none"> - 3 доступних платформ є лише веб сайт - Відсутньою декілька важливих функцій (особистий кабінет, система оплати) 	
Система Google Сайти		
Переваги	Недоліки	Ціна
<ul style="list-style-type: none"> - Повна та найбільша інтеграція з сервісами Google. (Google Drive, Google Maps та ін.) - Використання свого домену. - Використання дизайн системи Google. - Підтримка української версії. 	<ul style="list-style-type: none"> - Наявний дисковий простір у 15 Гб. - Неможливість використання сторонніх сервісів (тільки сервіси Google). - Немає підтримки користувальницьких CSS - Не можна використовувати Google Рекламу. 	Безкоштовний, проте можна замовити підтримку додаткових функцій від 144 грн на місяць.
Система Vigbo		
Переваги	Недоліки	Ціна
<ul style="list-style-type: none"> - Необмежений розмір дискового простору (за доплату) - Зручний інтерфейс конструктора сторінок. 	<ul style="list-style-type: none"> - Підтримка малої кількості сторонніх систем - Відсутня значна кількість важливих можливостей та 	Є наявність пробного періоду. Вартість від 160 грн на місяць.

	функцій. - Лише російська локалізація - З доступних платформ є лише веб сайт.	
Система Tilda		
Переваги	Недоліки	Ціна
- Найбільша кількість підтримуваних сторонніх сервісів. - Необмежений розмір дискового простору (за доплату).	- Обмежена кількість створюваних сторінок (500). - З доступних платформ є лише веб сайт. - Відсутність підтримки української мови. - Відсутній особистий кабінет користувача.	Є наявність пробного періоду, безкоштовного тарифу. Вартість від 195 грн на місяць.

Висновки

У результаті аналізу існуючих CMS систем, які можуть виступати, як аналог програмного забезпечення для динамічного представлення інформації, були виділені головні недоліки, що не дають змогу реалізувати дане завдання на їх основі: обмежений функціонал, перевантаженість систем непотрібним функціоналом та незручність використання, неможливість швидкого використання програмного забезпечення для інших користувачів, без дублювання всього коду програми або замовлення повторної послуги, що потребує додаткових коштів, неможливість доповнювати системи власним

функціональними модулями, при повільному інтернет з'єднанні завантаження сторінки займає великий проміжок часу, через універсальність коду, в багатьох системах відсутня українська локалізація.

Тому виникла необхідність в розробці власного програмного забезпечення на основі інформаційно-комунікаційної системи «ПОЛІДАР», що дає змогу динамічно та адаптивно представляти інформацію. При цьому система «ПОЛІДАР» дозволяє та надає можливість використовувати вже готові модулі, які полегшують розробку даного програмного забезпечення.

2 «ПОЛІДАР» – ПАРАЛЕЛЬНА ОБРОБКА ЛІНІЙНИХ ІРАЦІОНАЛЬНИХ ДАНИХ АЛГОРИТМАМИ РЕКОМБІНАЦІЇ

Інформаційно-комунікаційна система «ПОЛІДАР» – це програмний менеджер-органайзер, який керує та узгоджує роботу on-line доступних тематичних функціоналів, які по суті є окремими незалежними системами спеціалізованого призначення, що пов’язані між собою і функціонують як єдине ціле (тобто, як система), яких в свою чергу, може бути практично необмежена кількість. На даний час їх вже більше двадцяти і їхня кількість постійно збільшується. Їх кількість визначається кількістю розробок спеціалістами галузі алгоритмів потрібних їм інструментів чи окремих функціоналів, які програмуються та доповнюють Систему [9].

Структурна схема інформаційно-комунікаційної системи «ПОЛІДАР» показана на рис.2.1.

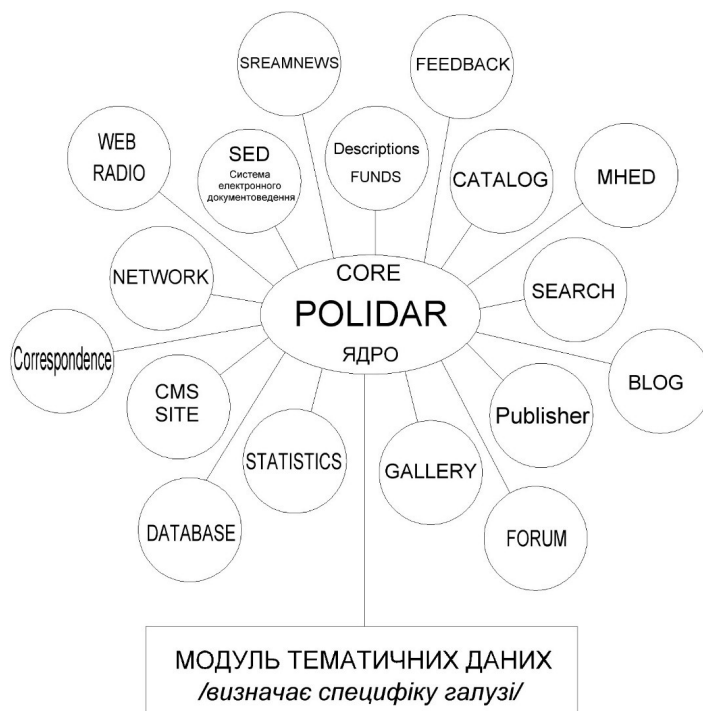


Рисунок 2.1 Інформаційно-комунікаційна система «ПОЛІДАР»

Основними функціоналами системи є:

CMS SITE – засіб для простого створення сайту;

NETWORK – функціонал, що забезпечує об'єднання сайтів в окрему мережу за допомогою порталу (потребує подальшої розробки);

SEARCH – системи пошуку різноманітного призначення, каталоги (довідники електронні);

Descriptions FUNDS – засіб для описання фондів, довідка на розташування фондів у пунктах збереження;

DATABASE – база даних (система, що забезпечує упорядковане зберігання інформації, даних, цифрових документів);

USER MODULE – модуль користувача, призначений для забезпечення у автоматичному режимі регламентованого доступу користувача до даних, цифрових документів;

GALLERY – галерея (система, яка призначена для відображення та створення багатoproфільних мультимедійних презентацій документів та перегляду їх on-line;

FORUM – функціонал для перегляду та створення тем для обговорення;

CHAT – функціонал для миттєвого обміну інформацією;

PLAN-REPORT – план-звіт (забезпечує створення планів-звітів в напівавтоматичному режимі);

WEB-RADIO – інтернет-радіо (система, що призначена для створення радіопередач та подальшої їх трансляції в ефірі, а в всесвітній павутині;

BLOG – діловий щоденник організації, система для фіксації поточного стану установи;

BACKUP – резервне копіювання контенту сайту або сайту цілком (функціонал, що дає можливість зберігати сайт на персональному комп'ютері);

STREAM AUDIO&VIDEO – функціонал, що дає можливість проводити аудіо та відео конференції;

ASSISTANCE – функціонал для дистанційної допомоги користувачам системи «ПОЛІДАР»;

SED – система електронного документоведення (на даний час забезпечує функціонування десять електронних журналів, які мають власні пошукові системи та багатoproфільні системи збирання і підрахунку статистичних даних);

MHED – система кодування даних для подальшої їх передачі загальним відкритим каналом зв'язку (всесвітньою мережею);

STREAMNEWS – інформаційний модуль (функціонал, який автоматично збирає новини з усіх сайтів архівних установ області та розміщує їх на порталі мережі);

Correspondence – система для листування в середині мережі, пересилання даних (дані можуть кодуватися за допомогою методу MHED /Multilayer Hybrid Encryption and Decryption/);

FEEDBACK – функціонал, що забезпечує зворотній зв'язок користувача з адміністрацією;

STATISTICS – система для глобального збирання відкритих статистичних даних про діяльність архівних установ всієї Київської області (функціонує в реальному часі), та ін.

Система «ПОЛІДАР» легко підлаштовується до різних вимоги користувача. Поєднує користувача та програмне забезпечення, яке в свою чергу знаходиться на VPS сервері.

Принцип роботи показаний на рис. 2.2.

Система «ПОЛІДАР» здатна підлаштовуватись практично до будь-яких необхідних вимог користувача.

Наприклад:

- Користувач №1. Користувач вже має свій власний розроблений сайт, який написав він сам, або сконструював за допомогою будь-якої CMS системи. В цьому випадку система «ПОЛІДАР» буде обслуговує даного користувача

таким чином. Користувачу буде запропоновано звертатися до системи, безпосередньо до Командера (Ядро). Командер надає з'єднання з хостингом, місцем, де буде зберігатися сайт користувача. Після цього відбудеться встановлення зв'язку «КОРИСТУВАЧ - ЯДРО-ХОСТИНГ».

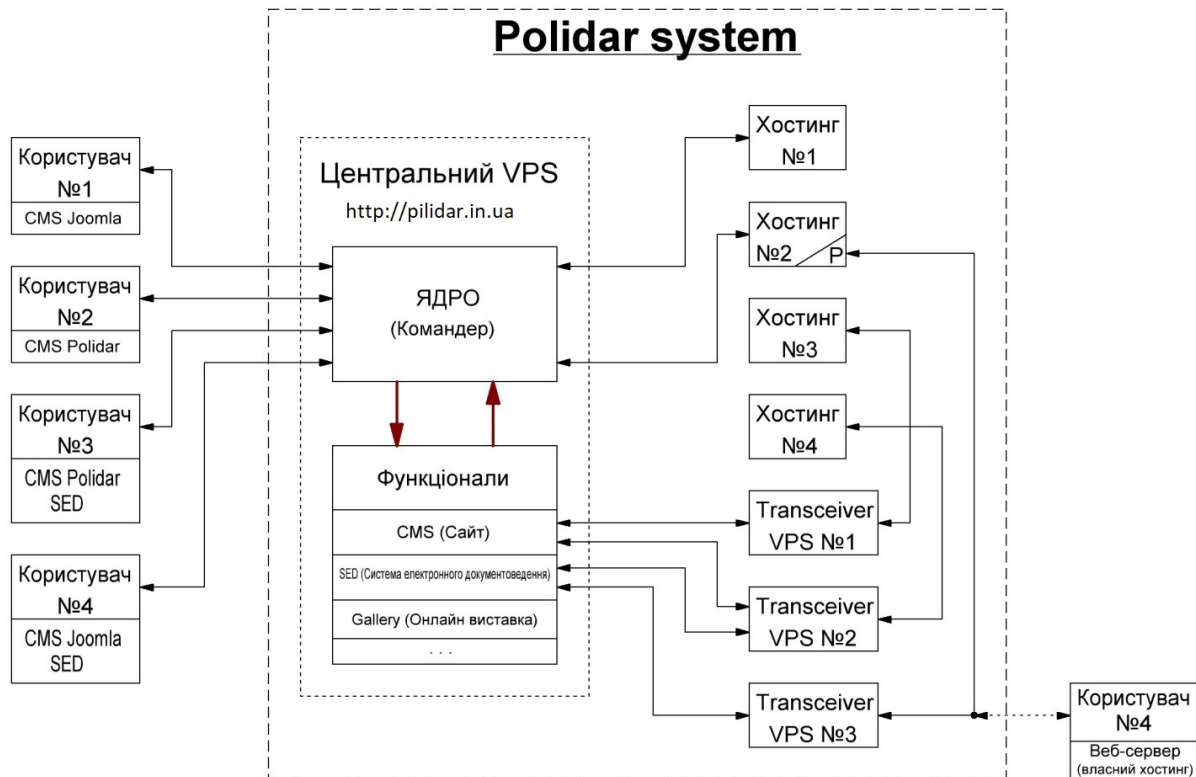


Рисунок 2.2 Схема, що пояснює принцип роботи системи

- Користувач №2. У користувача немає будь-якого сайту, та мінімальних навичок для його створення, але йому необхідний сайт. В цьому випадку «ПОЛІДАР» в автоматичному режимі створить новий сайт та обслуговуватиме користувача таким чином. Користувач надсилає звернення до системи, до Командеру (Ядро), Командер надає доступ до функціоналу «ПОЛІДАР», таким функціоналом є система CMS, та з'єднує з VPS Transceiver, на якому розміщуються додаткові компоненти функціоналу, що в свою чергу мають доступ до хостингу, де будуть зберігатись всі дані для сайту. Після цього відбудеться встановлення зв'язку «КОРИСТУВАЧ-ЯДРО-ФУНКЦІОНАЛ-ХОСТИНГ».

- Користувач №3. У користувача немає будь-якого сайту, та мінімальних навичок для його створення, але йому необхідний сайт, а також йому необхідний інструмент для документоведення SED. В цьому випадку «ПОЛІДАР» в автоматичному режимі створить та забезпечить доступ до SED, і буде обслуговувати користувача таким чином. Користувач надсилає звернення до системи, до Командеру (Ядро), Командер надає доступ до функціоналу «ПОЛІДАР» та забезпечую з'єднання з системою VPS Transceiver, де розташовані допоміжні системи та компоненти функціоналу, що в свою чергу мають доступ до хостингу, де будуть зберігатись всі дані для сайту та для інструменту документоведення. Після цього відбудеться встановлення зв'язку «КОРИСТУВАЧ-ЯДРО-ФУНКЦІОНАЛИ-ХОСТИНГ».
- Користувач №4. Користувач вже має повністю створений сайт, що написаний ним, або з використанням будь-якої CMS системи, а також він потребує інструменту документоведення SED. В цьому випадку «ПОЛІДАР» надасть доступ до системи SED, і обслуговуватиме користувача таким чином. Користувач надсилає звернення до системи, до Командеру (Ядро), Командер надає доступ до хостингу, місця, де будуть зберігатися всі данні сайту, також він зв'яже його з усім необхідним функціоналами «ПОЛІДАР» та забезпечує з'єднання з системою VPS Transceiver, де розташовані додаткові допоміжні системи та компоненти функціоналу, що в свою чергу мають безпосередній доступ до хостингу, де вже розміщені всі документи та данні цього сайту, також там зберігатимуться файли та дані інструменту документоведення. Після цього відбудеться встановлення зв'язку «КОРИСТУВАЧ-ЯДРО-ХОСТИНГ або ФУНКЦІОНАЛИ-ХОСТИНГ».

Висновки

Таким чином система «ПОЛІДАР» має достатню кількість гнучких функціоналів, які можуть задовільнити користувача з будь якими вимогами, і

ефективно використовуватися без допомоги висококваліфікованих комп'ютерних спеціалістів.

З СИСТЕМНИЙ ПІДХІД ДО РОЗРОБКИ СТРУКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Застосування системного підходу до створення програмного забезпечення для динамічного адаптивного представлення контенту дозволяє розширити область застосування, область використання сайту і не прив'язувати його структуру до певної галузі зі своєю специфікою роботи, швидко перелаштувати сайт та забезпечити легке та зручне управління контентом сайту, не орієнтувати структуру сайту на конкретну групу користувачів, а створювати його таким чином, що їм може скористатися практично кожен користувач своєї галузі.

ADAPTIVE CMS системи «ПОЛІДАР» містить в собі усі необхідні інструменти та надає весь функціонал для інтуїтивного та простого створення веб-сайту, що задовольняє сучасні вимоги:

- створення сайту з довільною структурою;
- можливість зручного та доступного керування вмістом (швидко та доступне створення постів та розділів, створення новини, галерей, переліку документів);
- можливість отримати данні про статистику відвідування сайту;
- можливість оцінити ефективність інформації користувачами;
- наявність сервісу безпосереднього зворотного зв'язку з адміністрацією сайту;
- забезпечення коректної роботи найбільших пошукових систем і алгоритмів та багато іншого.
- Наявність шаблонів сторінок для зручного та швидкого представлення інформації.

3.1 Постановка задачі

Необхідно розробити систему, що функціонує як динамічний адаптивний веб-сайт. Сайт має бути написаною з застосуванням таких мов веб-програмування: HTML, CSS, JavaScript PHP. Взаємодія та доступ сайту до бази даних MySQL має відбуватися за допомогою засобів та функцій веб-сервера Apache. Сайт має надавати інформацію суспільству, прямий обмін повідомленнями користувача та адміністратора установи. Має бути надана гарантована можливість перегляду веб-сайту з будь-якого пристрою, що підтримує останню версію сучасного браузеру.

3.2 Додаткові вимоги до веб-сайту

Виходячи з того, що в якості джерела подачі інформації для навчання було обрано інтернет-технології, то сформуємо ще деякі вимоги до даного веб-сайту:

- інтуїтивність в освоєнні, що не вимагатиме ніяких знань в області веб-програмування, достатнім має бути середній рівень користувача персонального комп'ютера;
- відсутність потреби у встановленні будь-яких додатків та програм для повного доступу до можливостей системи;
- сумісність з сучасними браузерами;
- адаптивність (можливість комфортного перегляду сайту на будь-якому екрані);
- динамічність (динамічна зміна контенту без додаткових перезавантажень);
- змога простої модифікації сайту з мінімальними залученням фінансів та інших витрат;
- забезпечення абсолютно автономної роботи для створення контенту на веб-сайті;

- сайт, як кінцевий продукт не має вимагати спеціальної додаткової доробки, проте він має таку можливість;

3.3 Вимоги до оформлення матеріалів, які викладаються на сайті

Може бути обрана будь-яка зручна форма подання матеріалу користувачам. Це можуть бути фотографії, графіки, малюнки, текст і т.д. Все це може бути розташоване на сторінці в довільній формі. При поданні інформації, ця сама інформація має легко та доступно сприйматися. Сторінка не повинна містити зайвої або перевантаженої інформації (тесту або графіків), яка б могла відвернути увагу користувача від головної інформації.

Усі матеріали мають бути чітко структуровані та легкодоступні. Матеріали можуть передаватися особі, будь-яким способом. Це може бути і безпосередня завантаження і читання документа з веб-сервера. Адаптивність в свою чергу має забезпечувати зручність та доступність сприйняття інформації практично з будь-якого пристрою (персональний комп'ютер, ноутбук, планшет, мобільний телефон та інше).

Висновки

Системний підхід до дослідження поставленої задачі дозволив виробити основні та додаткові вимоги до програмного забезпечення динамічного адаптивного представлення контенту, а також вимоги до матеріалів, які викладаються на сайті.

4 РОЗРОБКА СТРУКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Відповідно до висунених в розділі 3 вимог необхідно сформувати структуру програмного забезпечення. Для цього розробимо інфологічну та датологічну моделі предметної області, виберемо програмні засоби, сервер баз даних, мову написання сценаріїв, створимо архітектуру програмного засобу, розробимо файлову структуру продукту та розробимо алгоритм зворотного зв'язку.

4.1 Опис принципу роботи та функціоналу використаних інструментів

При розробці програмного забезпечення для забезпечення швидкодії, простоти використання, легкості впровадження продукту та подальшої його модифікації були використані наступні інструменти розробки: Laravel, MVC, Vuejs, Vuejs React, Webpack. Розглянемо кожний з них.

Laravel - це безкоштовний веб-фреймворк PHP з відкритим кодом, створений Тейлором Отвеллом і призначений для розробки веб-додатків за архітектурною схемою модель-перегляд-контролер (MVC) та заснована на Symfony. Деякі особливості Laravel - це модульна система упаковки з виділеним менеджером залежностей, різні способи доступу до реляційних баз даних, утиліти, які допомагають у розгортанні та обслуговуванні додатків, та його орієнтація на синтаксичний цукор. Основною перевагою Laravel є зручна маршрутизація продукту, немає необхідності створювати окремі html файли для кожної сторінки та будувати складну файлову систему продукту. Це все дає змогу спростити логіку маршрутизації продукту та звільняє від необхідності дублювати однаковий код програми.

Контролер моделі перегляду (зазвичай відомий як MVC) - це модель дизайну програмного забезпечення, яка зазвичай використовується для

розробки інтерфейсів користувача, яка розділяє відповідну логіку програми на три взаємопов'язані елементи. Це робиться для того, щоб відокремити внутрішнє представлення інформації від способів подання та прийняття інформації від користувача. Цей вид патерну використовується для проектування макета сторінки. Нижче наведена структура взаємодії компонентів [10].

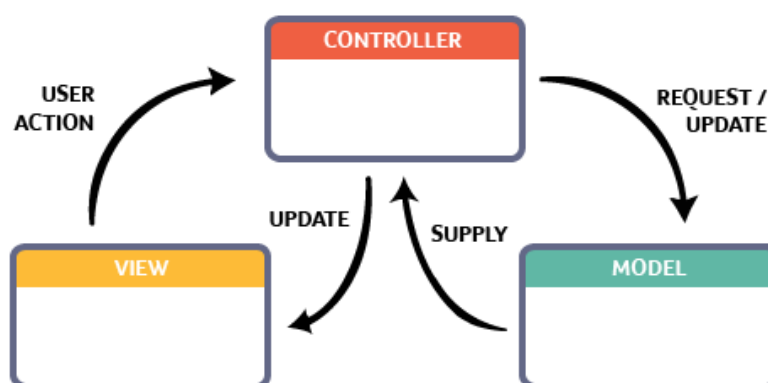


Рисунок 4.1.1 – Діаграма взаємодії між компонентами шаблону

Традиційно використовуючись для графічних інтерфейсів користувачів настільних ПК (GUI), ця модель стала популярною для розробки веб-додатків. Популярні мови програмування, такі як JavaScript, Python, Ruby, PHP, Java, C # і Swift, мають фреймворки MVC, які використовуються для розробки Інтернету або мобільних додатків «прямо з коробки».

Vue.js - це фреймворк JavaScript з відкритим кодом для моделювання інтерфейсу користувача та односторінкових програм з відкритим кодом. Він був створений Evan You, і підтримується ним та рештою активних членів основної команди, що надходять від різних компаній, таких як Netlify та Netguru.

Основною можливістю фреймворку є створення односторінкових продуктів. Продукти, створені за допомогою Vue.js, за своїм функціоналом повністю відповідають традиційним багатосторінковим продуктам. Проте

для користувача все відбувається в межах однієї сторінки, контент на якій завантажується та відображається динамічно.

Нижче (рис. 4.1.2) наведена схема життєвого циклу примірника (instance) [11].

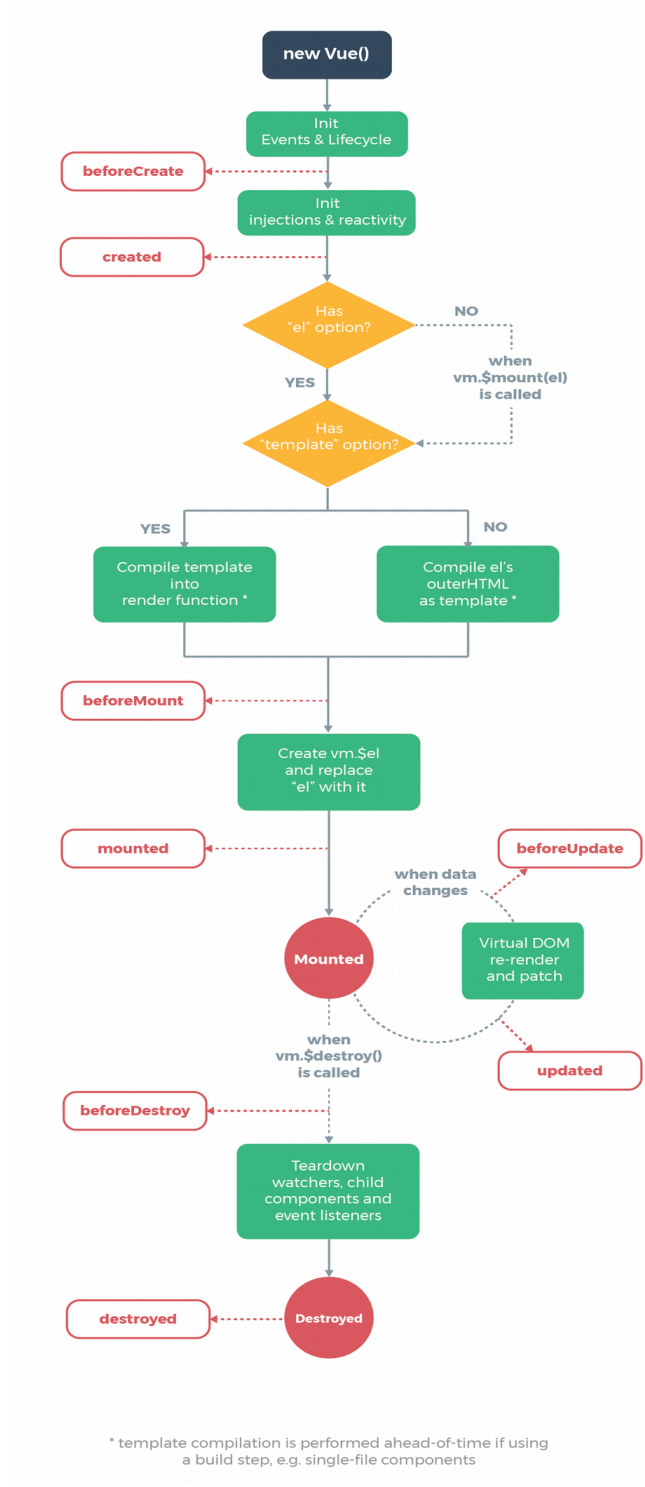


Рисунок 4.1.2 – Діаграма життєвого циклу Vuejs

Реактивність (Vuejs React). Однією з найбільш відмітних особливостей Vue є ненав'язлива система реактивності. Моделі - це просто звичайні об'єкти JavaScript. Коли ви їх модифікуєте, представлення оновлюється. Це робить управління проектом простим та інтуїтивним. Коли передається звичайний об'єкт JavaScript в екземпляр Vue як його параметр даних, Vue буде проходити всі його властивості та перетворювати їх в getter / setters за допомогою Object.defineProperty. Геттер / сеттер невидимий для користувача, але «під кришкою» вони дозволяють Vue здійснювати відстеження залежностей. Опис слідування за зміною стану компоненту наведено нижче [12].

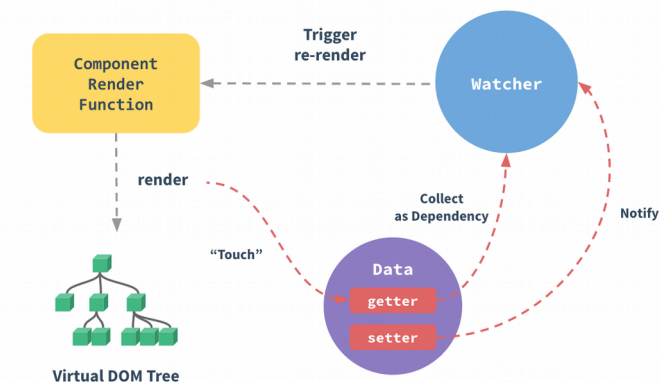


Рисунок 4.1.3 – Діаграма слідування за змінами

Vue.js - один з найпопулярніших фреймворків JavaScript, і його підтримують багато веб- та гібридно-мобільних розробників для своїх особистих та робочих проектів.

Vuetify розроблений саме відповідно до специфікації Material Design. Кожен компонент створений вручну. Розробка не зупиняється на основних компонентах, викладених у специфікаціях Google. Завдяки підтримці членів громади та спонсорів розробляються доступні додаткові компоненти.

Vuetify будується з нуля, щоб бути семантичним. Легкий у навчанні, легко запам'ятовується. Vuetify має послідовний цикл оновлень, що дозволяє частіше отримувати виправлення помилок та вдосконалення.

Vuetify постійно оновлюється, реагуючи на проблеми та запити спільноти з випередженням. Опис дисплеїв, що підлягають адаптивності наведено нижче [13].






Material Design Viewport Breakpoints			
Device	Code	Types	Range
 Extra small	xs	small to large handset	< 600px
 Small	sm	small to medium tablet	600px > < 960px
 Medium	md	large tablet to laptop	960px > < 1264px*
 Large	lg	desktop	1264px* > < 1904px*
 Extra large	xl	4k and ultra-wides	> 1904px*
* -16px on Desktop			

Рисунок 4.1.4 – Розміри екранів для керування адаптивністю

Webpack - це постачальник модулів JavaScript з відкритим кодом. Це модульний постачальник, насамперед, для JavaScript, але він може трансформувати активні елементи, такі як HTML, CSS та зображення, якщо включені відповідні завантажувачі. Webpack приймає модулі із залежностями та генерує статичні активи, що представляють ці модулі.

Webpack приймає залежності та генерує графік залежності, що дозволяє веб-розробникам використовувати модульний підхід для цілей розробки веб-додатків. Він може бути використаний з командного рядка, або може бути налаштований за допомогою конфігураційного файлу, який називається `webpack.config.js`. Цей файл використовується для визначення правил, плагінів тощо для проекту. Для своєї роботи Webpack потребує Node.js.

4.2 Побудова інфологічної моделі предметної області

Інфологічна модель предметної області – це опис предметної області, виконаний з використанням спеціальних мовних засобів, що не залежать від використання надалі програмних засобів. Вимоги до інфологічної моделі:

- адекватність відображення предметної області;
- несуперечливість;
- відображення потреби всіх користувачів майбутньої інформаційної системи;
- допущення можливих розширень моделі, а також при великій розмірності інфологічної моделі;
- композиції і декомпозиції, бажано із застосування мови опису предметної області як при ручному, так і при автоматизованому проектуванні;
- легкість сприйняття різними категоріями користувачів.

Інфологічна модель є ядром системи проектування. Вона містить всю необхідну інформацію для проектування програми.

Для опису інфологічної моделі даних можна виділити дев'ять об'єктів, які беруть участь в обробці інформації:

- "Сторінка відображення інформації";
- "Службова сторінка";
- "Меню";
- "Конструктор пункту меню";
 - "Проста сторінка"
 - "Проста сторінка плюс"
 - "Посилання на зовнішній ресурс"
 - "Шаблони"
 - "Контакти"
 - "Список сторінок"
 - "Сторінка додаткового меню"

- "Перелік персоналу"
- "Список документів"
- "Конструктор блоку відображення «Header»";
- "Налаштування"

Кожен об'єкт має свої властивості:

Об'єкт " Сторінка відображення інформації " являє собою сторінку відображення контенту. Безпосередньо цю сторінку бачить відвідувач продукту. Данна сторінка є адаптивною, що дає змогу зручно переглядати контент практично на будь-якому пристрої, також сторінка є динамічною, це в свою чергу забезпечує користування продуктом без багаторазових перезавантажень сторінки. Даний об'єкт складається з декількох динамічних адаптивних блоків. А саме. Блок "Header", цей блок знаходиться в самій верхній точці сторінки. Він призначений для відображення загальної інформації про продукт. Може містити рядки з текстом, логотипи, зображення та інше. Блок «Меню», даний блок має можливість бути розташованим в декількох місцях (праворуч, ліворуч або вгорі сторінки). Цей блок являє собою навігаційне меню всього продукту. З його допомогою можна зручно знаходити необхідну інформацію. Найбільшим полем є «Поле Контенту», завданням цього поля є виконання основної функції програмного забезпечення, а саме динамічне та адаптивне відображення контенту. Цей блок може приймати різноманітний вигляд, який залежить від типу контенту, який необхідно представити.

Об'єкт " Службова сторінка " призначена для адміністрації продукту. Вона дає змогу фактично створити сам продукт з нуля. Також вона забезпечує легке керування над всіма аспектами продукту.

Об'єкт " Меню " дає змогу зручно та швидко створювати меню продукту (всю його структуру). Також на цій сторінці відбувається редагування вже створеного меню, а саме зміна назви пункту меню, зміна

його відносного розташування (підменю), перехід до конструктора вмісту пункту меню.

Об'єкт " Конструктор пункту меню " – це основний інструмент для створення сторінок на продукті. Дає змогу зручно та швидко створювати сторінки для представлення інформації у різноманітних формах.

" Проста сторінка " – дає змогу створювати сторінки, які містять в собі заголовок та текст.

" Проста сторінка плюс " – копіює функціонал «Простої сторінки» та додатково дає змогу створювати анотацію до тексту. Дана анотація використовується при відображенні даної сторінки в списку. Це дає змогу користувачу попередньо ознайомитись з вмістом сторінки без переходу на неї.

" Посилання на зовнішній ресурс " – створює сторінка, при переході на яку буде здійснюватися переадресація на обраний зовнішній ресурс.

" Шаблони " – дає змогу створювати сторінки з наявних шаблонів, які краще підходять для відображення того чи іншого типу інформації.

Шаблон " Контакти " – шаблон за допомогою якого можна створити сторінку для красивого та інформативного відображення адреси, графіку роботи та іншої інформації даного типу.

Шаблон " Список сторінок " – шаблон за допомогою якого можна створити сторінку для красивого та інформативного відображення списку вже створених сторінок, також шаблон дає можливість налаштувати параметри відображення списку (кількість одночасно представлених сторінок, їх порядок та інше).

Шаблон " Сторінка додаткового меню " – шаблон за допомогою якого можна створити сторінку для відображення додаткових опцій та дій для даної сторінки.

Шаблон "Перелік персоналу" – шаблон за допомогою якого можна створити сторінку для красивого та інформативного відображення переліку персоналу з фотографіями та невеликим описовим текстом.

Шаблон "Список документів" – шаблон за допомогою якого можна створити сторінку для красивого та інформативного відображення переліку документів.

Об'єкт "Конструктор блоку відображення «Header»" дає змогу створювати головний банер продукту. Може мати різну структуру для відображення таких типів даних як текст, фото, логотип та інше.

Об'єкт "Налаштування" призначений для зручного керування та налаштування параметрів продукту.

На рисунку 4.2.1 зображена блок схема, що відображає структуру і взаємозв'язок сторінок налаштування та головної сторінки.

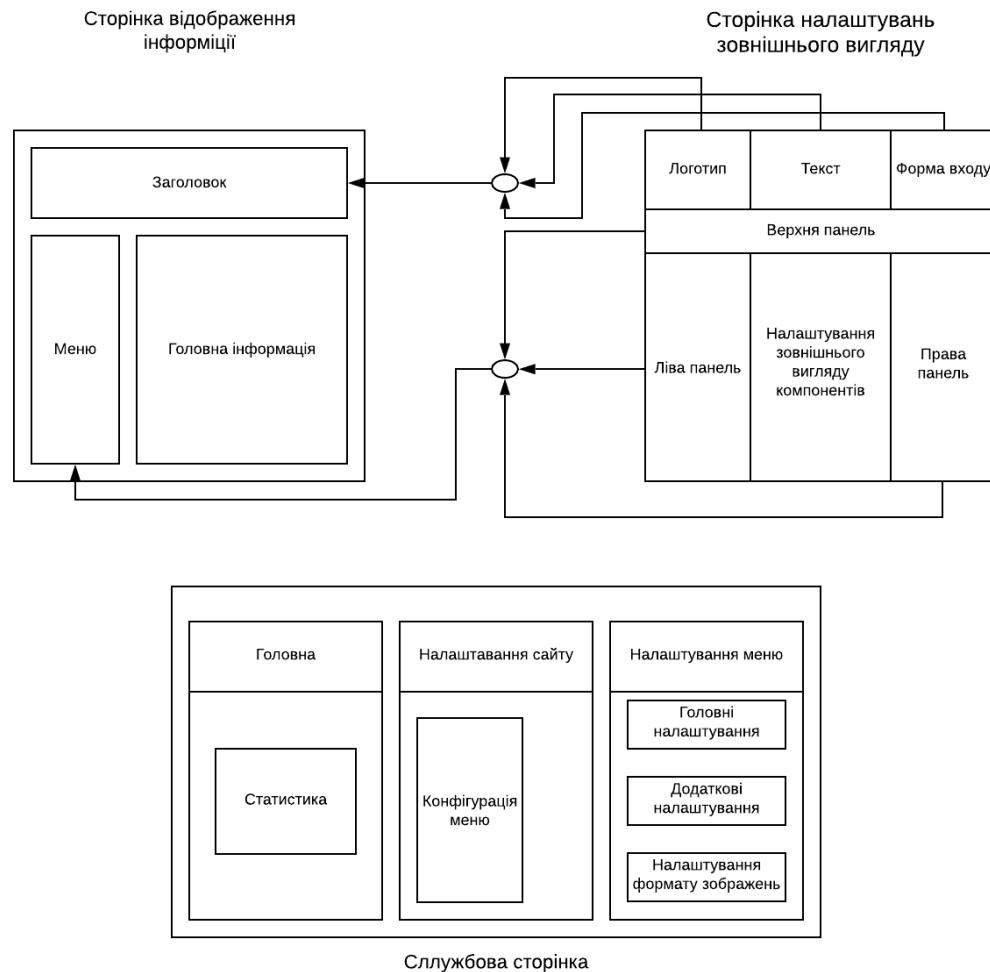


Рисунок 4.2.1 – Структура програмного забезпечення

4.3 Побудова даталогічної моделі предметної області

На підставі аналізу побудованої інфологічної моделі була розроблена структура бази даних. Проектування бази даних полягає в тому, щоб сформувати ряд таблиць з даними. Реляція бази даних з заданої фізичної структури полягає в тому, щоб розбити необхідну всю інформацію на таблиці, а також створити необхідну кількість полів для кожної з таблиць і встановити взаємозв'язки між кожною з цих таблиць. Особливістю проектування є необхідність мінімізації небажаних таблиць та уникання дублювання інформації в них. Також важливим аспектом є створення оптимальної мінімальної кількості інформаційних полів в кожній з цих

таблиць. У кожній таблиці є поле, що містить в собі унікальний ідентифікаційний номер, цей номер дає змогу однозначно та безпомилково ідентифікувати запис в цій таблиці. Така структура виключає ймовірність існування надмірності даних, дозволяє виконувати вибір даних таблиці за умовою, унеможлиблює ймовірність збоїв та конфліктів при одночасній роботі декількох користувачів.

Однією з основних таблиць є таблиця "menus". У цій таблиці зберігається вся інформація про навігаційне меню створюваного продукту.

Розглянемо поля, що наявні в цій таблиці.

- Поле "id". Використовується для послідовної унікальної нумерації даних в таблиці. При додаванні нового елементу до таблиці дане поле автоматично заповниться потрібним (наступним, унікальним) номером. Цей номер дає змогу в подальшому зручно ідентифікувати пункт меню.
- Поле "parent_id". Якщо даний елемент меню є підменю, тобто має певний батьківський пункт меню, то в це поле записується унікальний номер батьківського пункту меню. Якщо елемент не є підменю, то в дане поле записується нуль. Така система дає змогу створювати меню з необмеженою кількістю підменю.
- Поле " site_id". Оскільки програмне забезпечення надає змогу одному користувачу створювати декілька продуктів, то потрібно зазначити який елемент меню до якого продукту належить. Цю функцію і виконує дане поле. Перевагою цього є відсутність необхідності створювати нову таблицю для кожного продукту.
- Поле " sort". В цьому полі зазначається порядок відображення пунктів меню.
- Поле " title". Це поле містить назву пункту меню.

- Поле " link". Дане поле використовується, якщо користувач хоче аби при переході в це меню відбувалася переадресація на зовнішній ресурс. Тут зберігається url адреса необхідного ресурсу.
- Поле " post_id". Містить ідентифікатор посту, який має відображатися при переході на цей пункт меню.
- Поле " category_id". Містить ідентифікатор категорії (списку постів)
- Поле " type_link". Містить інформацію про тип відображеної сторінки (посилання на зовнішній ресурс, категорія або сторінка з контентом)
- Поле " status". Відображає статус елемента меню. Дає можливість створювати елементи та не публікувати їх одразу.
- Поле " created_at". Тут зазначається дата і час створення елемента меню.
- Поле " updated_at". Тут зазначається дата і час останнього оновлення елемента меню.

Типи даних полів таблиці " menus " представлені в табл.4.3.1.

Таблиця 4.3.1 – Структура таблиці " menus "

№	Опис	Поле	Тип даних	Додаткові параметри
1	Ідентифікатор	id	int(11)	AUTO_INCREMENT
2	Ідентифікатор батьківського елемента	parent_id	int(11)	'0'
3	Ідентифікатор продукту	site_id	int(11)	NULL
4	Сортування	sort	int(11)	NULL
5	Заголовок	title	varchar(50)	NULL
6	Посилання	link	Mediumtext(16777215)	NULL

7	Ідентифікатор посту	post_id	Mediumtext(16777215)	NULL
8	Ідентифікатор категорії	category_id	Mediumtext(16777215)	NULL
9	Тип сторінки	type_link	Mediumtext(16777215)	NULL
10	Статус	status	int(11)	NULL
11	Дата створення	created_at	TIMESTAMP	NULL
12	Дата останнього оновлення	updated_at	TIMESTAMP	NULL

В табл.4.3.2 " sites " зберігаються всі відомості про продукти.

Унікальними полями в даній таблиці є наступні поля.

- Поле "user_id". Це поле вказує користувача, який створив даний продукт.
- Поле " alias". Дане поле вказує назву продукту, що буде відображатися в url адресі.
- Поле " name". Вказує внутрішню назву продукту.
- Поле " title". Використовується для задання заголовку html сторінки.
- Поле " type". Використовується для вказання того чи активований продукт.

Таблиця 4.3.2 – Структура таблиці " sites "

№	Опис	Поле	Тип даних	Додаткові параметри
1	Ідентифікатор	id	int(11)	AUTO_INCREMENT
2	Ідентифікатор	user_id	int(11)	NULL

	користувача			
3	URL адреса	alias	varchar(50)	NULL
4	Назва	name	varchar(250)	NULL
5	Заголовок	title	varchar(250)	NULL
6	Активація	type	varchar(50)	NULL
7	Дата створення	created_at	TIMESTAMP	NULL
8	Дата останнього оновлення	updated_at	TIMESTAMP	NULL

В табл.4.3.3 " posts " зберігаються всі відомості про пости.

Унікальними полями в даній таблиці є наступні поля.

- Поле " title". Дане поле вказує заголовок посту.
- Поле " title ". Зберігає стислий опис посту для відображення в анотації.
- Поле " text ". Зберігає текст посту.
- Поле " type ". Вказує тип посту. Може бути обраним з наявних шаблонів.
- Поле " comment_status ". Вказує чи наявні коментарі на цій сторінці.
- Поле " status ". Вказує чи відображається дана сторінка
- Поле " like ". Вказує чи наявні вподобання на цій сторінці.
- Поле " show". Кількість переглядів даного поста.
- Поле " template". Зберігає данні про зовнішній вигляд сторінки.
- Поле " files". Зберігає файли, що доступні на цій сторінці.
- Поле " time_publication". Вказує час відкладеного публікування.

Таблиця 4.3.3 – Структура таблиці " posts "

№	Опис	Поле	Тип даних	Додаткові параметри
1	Ідентифікатор	id	int(11)	AUTO_INCREMENT

	Р			Т
2	Ідентифікатор продукту	site_id	int(11)	NULL
3	Ідентифікатор користувача	user_id	int(11)	NULL
4	Заголовок	title	varchar(50)	NULL
5	Анотація	annotation	text (65535)	NULL
6	Текст	text	text (65535)	NULL
7	Тип	type	varchar(50)	NULL
8	Коментарі	comment_status	int(11)	'0'
9	Статус	status	int(11)	NULL
10	Вподобання	like	int(11)	NULL
11	Відображення	show	int(11)	NULL
12	Зовнішній вигляд	template	JSON	NULL
13	Файли	files	JSON	NULL
14	Дата публікації	time_publication	TIMESTAMP	NULL
15	Дата створення	created_at	TIMESTAMP	NULL
16	Дата останнього оновлення	updated_at	TIMESTAMP	NULL

Табл.4.3.4 " options" зберігаються всі відомості про пости.

Унікальними полями в даній таблиці є наступні поля.

- Site_id вказує до якого продукту відносяться дані параметри
- Metatag поле для запису атрибутів або параметрів
- Value поле для запису значень атрибутів або параметрів

Таблиця 4.3.4 – Структура таблиці " options"

№	Опис	Поле	Тип даних	Додаткові параметри
1	Статус	Site_id	int(11)	NULL
2	Метатег	Metatag	varchar(50)	NULL
3	Запис атрибутів або параметрів	Value	varchar(50)	NULL

Табл.4.3.5 " headers" зберігаються всі відомості про пости.

Унікальними полями в даній таблиці є наступні поля.

- Поле " template". Зберігає зовнішній вигляд заголовку продукту.

Таблиця 4.3.5 – Структура таблиці " headers"

№	Опис	Поле	Тип даних	Додаткові параметри
1	Ідентифікатор	id	int(11)	AUTO_INCREMENT
2	Ідентифікатор продукту	site_id	int(11)	'0'
3	Зовнішній вигляд	templat e	JSON	NULL

Табл.4.3.6 " category_posts" зберігаються всі відомості про пости.

Унікальними полями в даній таблиці є наступні поля.

- Поле " category_id". Зберігає ідентифікатор категорії.
- Поле " post_id". Зберігає ідентифікатор посту.

Таблиця 4.3.6 – Структура таблиці " category_posts"

№	Опис	Поле	Тип даних	Додаткові параметри
---	------	------	-----------	---------------------

1	Ідентифікатор категорії	category_id	int(11)	NULL
2	Ідентифікатор посту	post_id	int(11)	NULL

Табл.4.3.7 "categories" зберігаються всі відомості про пости.

Унікальними полями в даній таблиці є наступні поля.

- Поле " name". Зберігає інформацію категорії.

Таблиця 4.3.7 – Структура таблиці " categories"

№	Опис	Поле	Тип даних	Додаткові параметри
1	Ідентифікатор	id	int(50)	AUTO_INCREMENT
2	Назва контейнеру	name	varchar(50)	NOT NULL
3	Дата останнього оновлення	updated_at	TIMESTAMP	NULL
4	Дата створення	created_at	TIMESTAMP	NULL

На рисунку 4.3.1 наведена структура бази даних з вказаними взаємозв'язками відповідних пунктів таблиць. Діаграму було створено за допомогою сервісу dbdiagram.io [14].

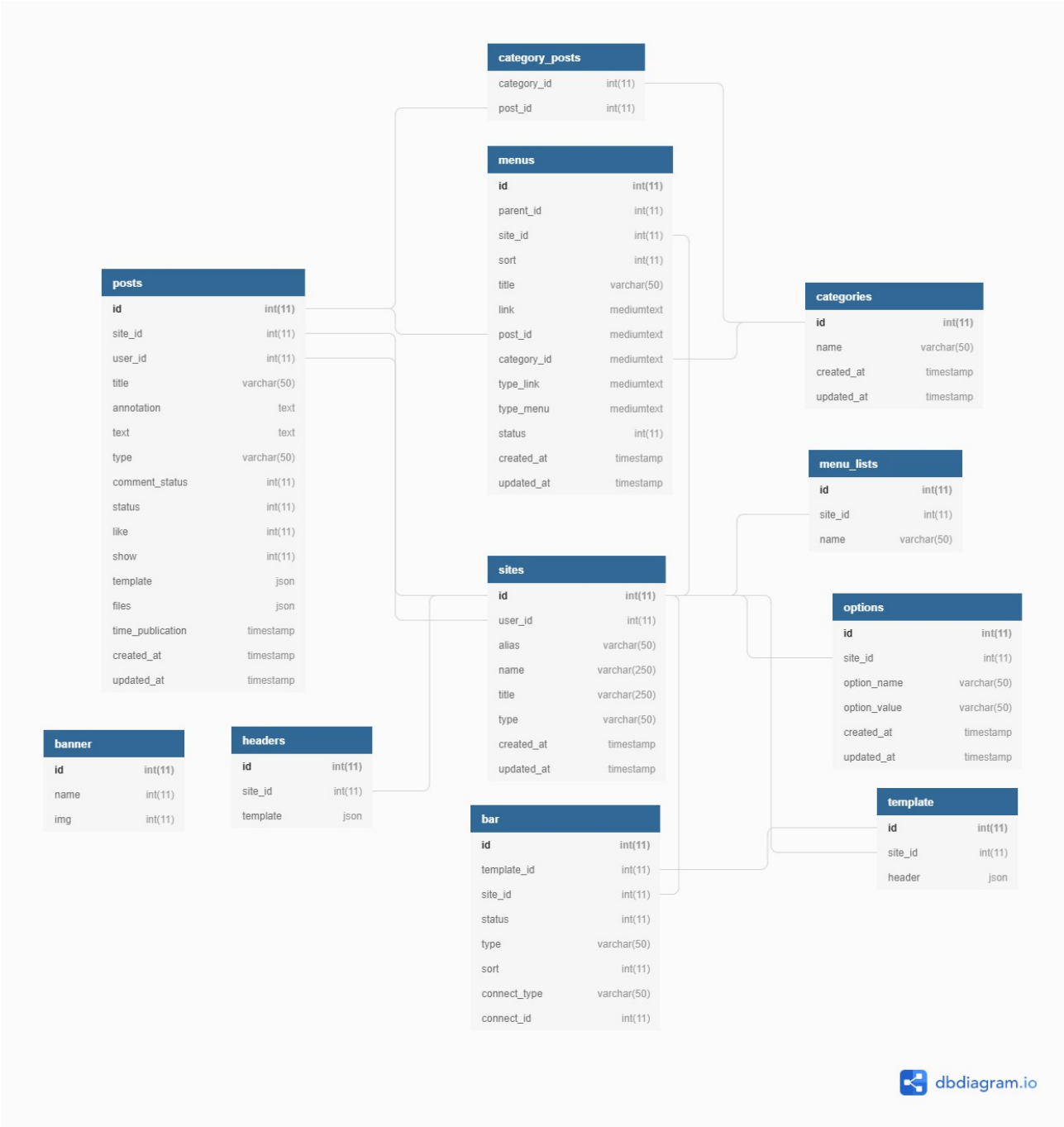


Рисунок 4.3.1 – Структурна схема бази даних

4.4 Розробка сайту програмного забезпечення

Програмне забезпечення було вирішено розробляти на основі веб-сайту. Це обумовлено тим, що доступ до продукту отримає максимальна кількість користувачів, адже немає потреби завантажувати та встановлювати окремий

додаток на свій пристрій. Достатньо зайти на веб-сайт практично з будь-якого пристрою, що підтримує останні версії веб-браузерів.

Основною задачею при його розробці було створити сайт, котрим буде легко та доступно користуватись на пристрої практично з будь-якою формою дисплею. Також необхідно розробити сам інтерфейс. Він має бути зручним, не перевантаженим надмірним функціоналом проте одночасно надавати всю необхідну інформацію в доступній та зрозумілій формі, не в останню чергу мати приємний та естетичний вигляд.

Програмне забезпечення загалом складається з двох сторінок: головної сторінки, яка є публічною та безпосередньо призначена для динамічного адаптивного відображення контенту та сторінка налаштувань, доступ до якої мають лише адміністратори продукту, на цій сторінці відбувається налаштування всього продукту загалом, а саме створення розділів, навігаційного меню, налаштування контенту (кількість та місце відображення), обрання кольорової гамми, задання заголовку (назви) продукту та інше.

4.4.1 Сторінка налаштування зовнішнього вигляду

Данна сторінка (рис. 4.4.1a) призначена для конфігурації вигляду веб-сторінки, яку буде бачити безпосередньо користувач програмного забезпечення.

Ця сторінка складається з двох частин. Перша це перед показ сторінки, що створюється. Друга це інструментарій налаштування, в якому задаються всі необхідні параметри та властивості компонентів сайту.

Сторінка складається з таких компонентів:

- Головна комірка
- Верхня панель
- Ліва панель

- Права панель

Налаштування головної комірки відбувається наступним чином:

- Обирається кількість компонентів з яких буде складатися головна комірка (один, два або три).
- Задається розмір відповідних компонентів.
- Обирається елемент, що буде налаштовуватися (лівий, центральний, правий).
- Визначається тип контенту, що буде відображатися (логотип, текст, форма входу).
- Відбувається безпосереднє налаштування обраного типу елементу (завантаження логотипу, написання заголовку та його конфігурація і т.д.)
- В разі необхідності обирається колір тла або завантажуються картинка, для відображення її на фоні.

Налаштування панелей відбувається наступним чином:

- Задається параметр відображення елементу (присутній чи відсутній на сторінці).
- Обирається кількість стільників з інформацією. Подальшому налаштуванні в ці комірки можна вставити різні типи контенту (звичайний текст, посилання, банер, навігаційне меню і т.д.).
- Обирається колір тла панелі.

На рисунку 4.4.1б наведено приклад оформлення сторінки.

На рисунку 4.4.1в наведено алгоритм налаштування головної комірки.

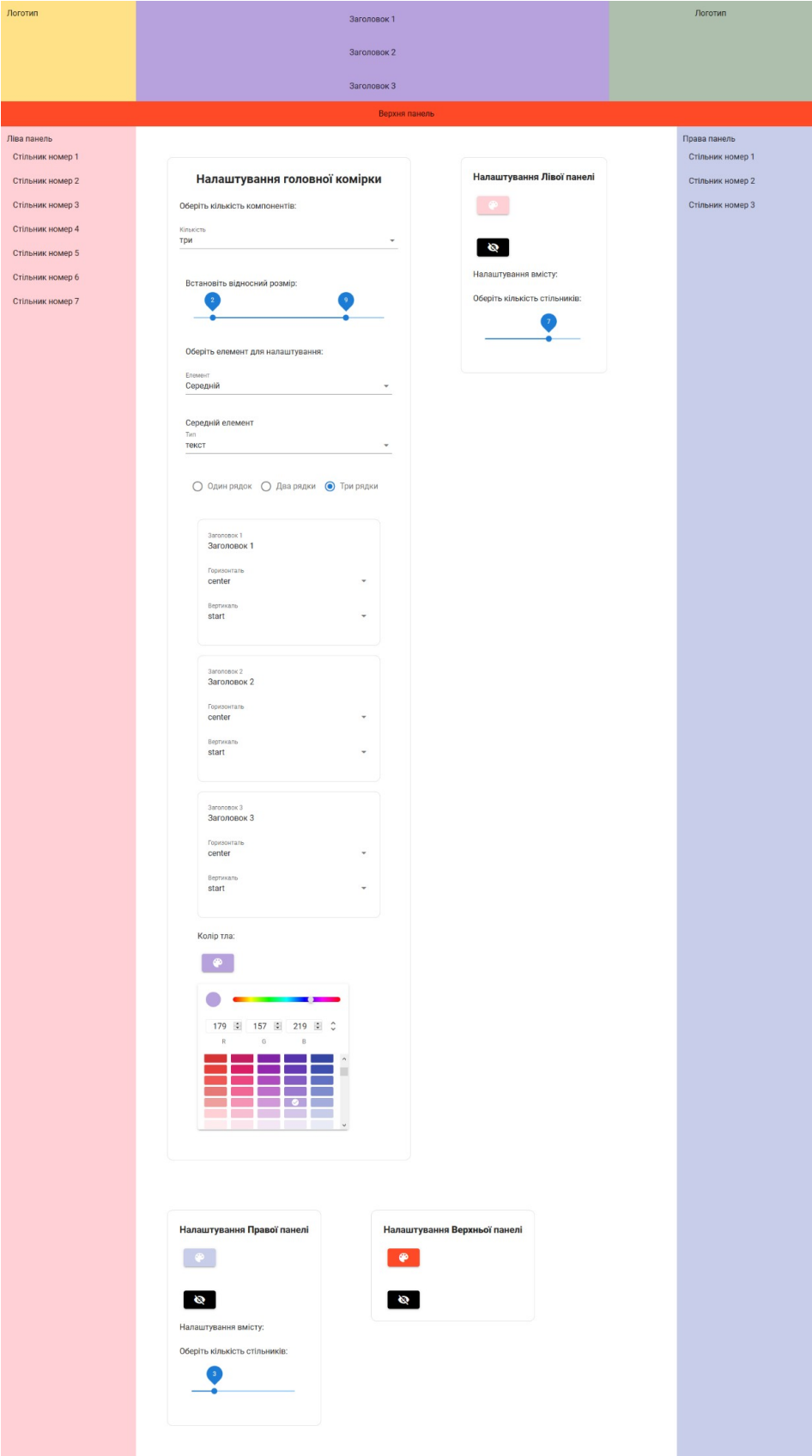


Рисунок 4.4.1a – Сторінка налаштувань зовнішнього вигляду



Рисунок 4.4.16 – Приклад оформлення сторінки

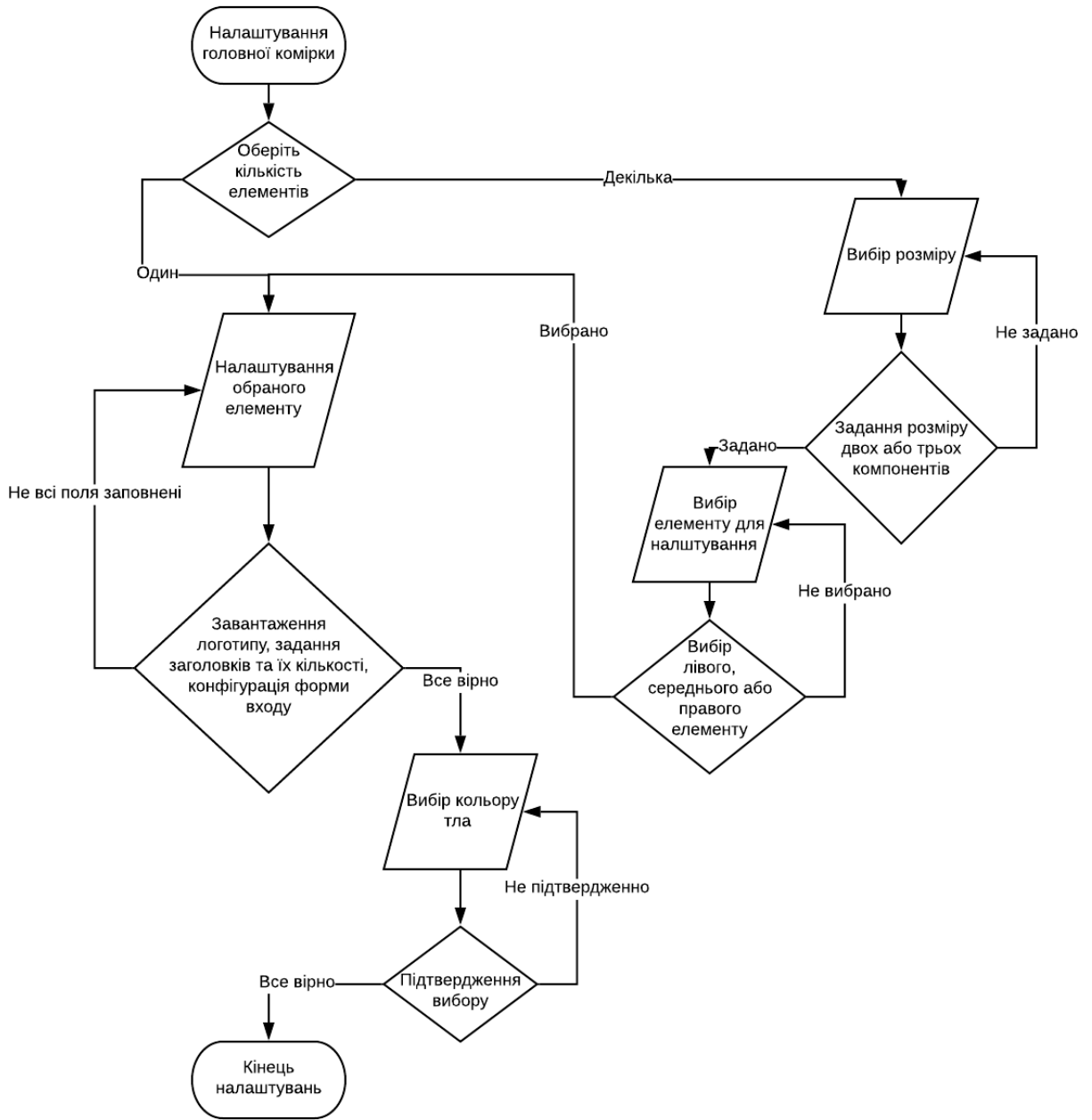


Рисунок 4.4.1в – Алгоритм налаштування головної комірки

4.4.2 Головний розділ службової сторінки

Даний розділ (рис. 4.4.2) призначений для відображення статистики робочого сайту. А саме загальна кількість відвідувань сайту за обраний проміжок часу, кількість переглядів обраної статті, кількість завантажень файлу і таке інше.

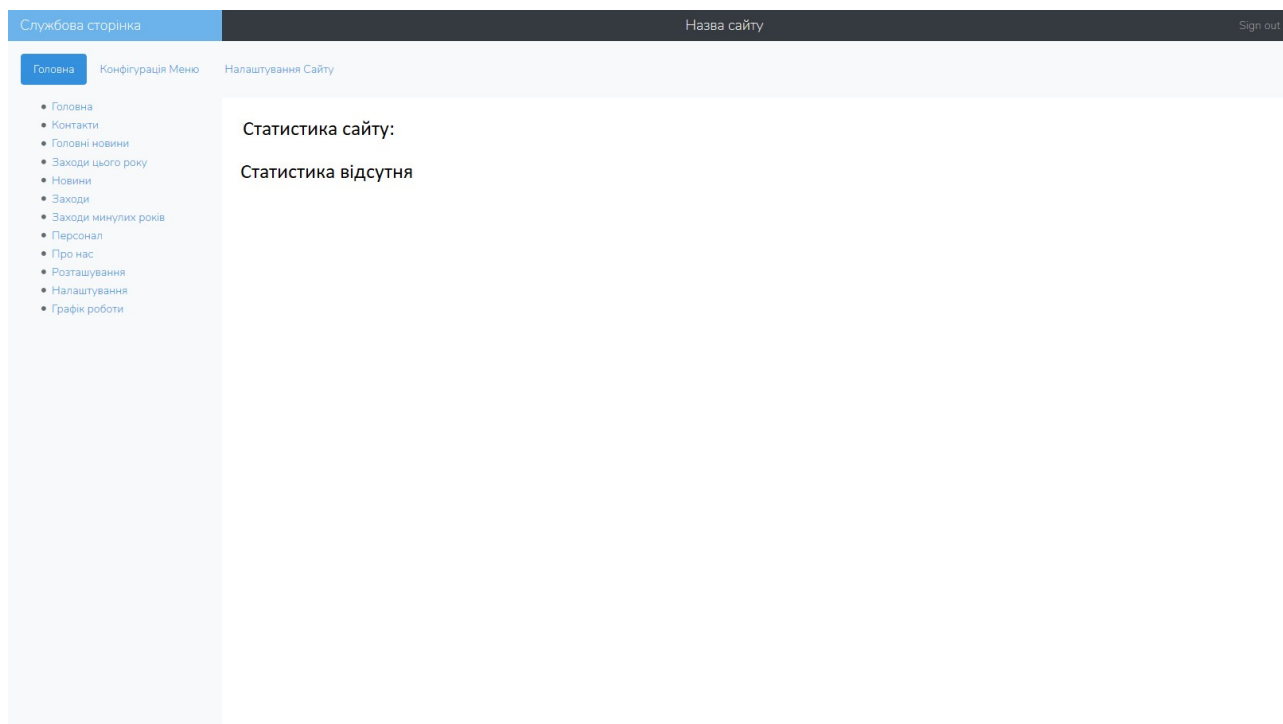


Рисунок 4.4.2 – Головний розділ сторінки налаштувань

4.4.3 Розділ конфігурації меню службової сторінки

Даний розділ (рис 4.4.3) призначений для створення та налаштування меню сайту.

Налаштування меню відбувається наступним чином:

- В поле вводу вписується назва бажаного розділу меню та натискається кнопка «Створити новий розділ». Після цього на початку всього меню створиться новий розділ.
- Далі за допомогою миші можливо розміщувати розділи у необхідному порядку. Також можливо встановлювати певні розділи як підрозділи інший. Ступінь вкладеності підрозділів є необмеженою.

- За необхідності при натисканні кнопки «Налаштування» біля необхідного розділу можна перейти до вікна конфігурації (рис 4.4.4), в якому можливо змінити назву розділу або видалити його.

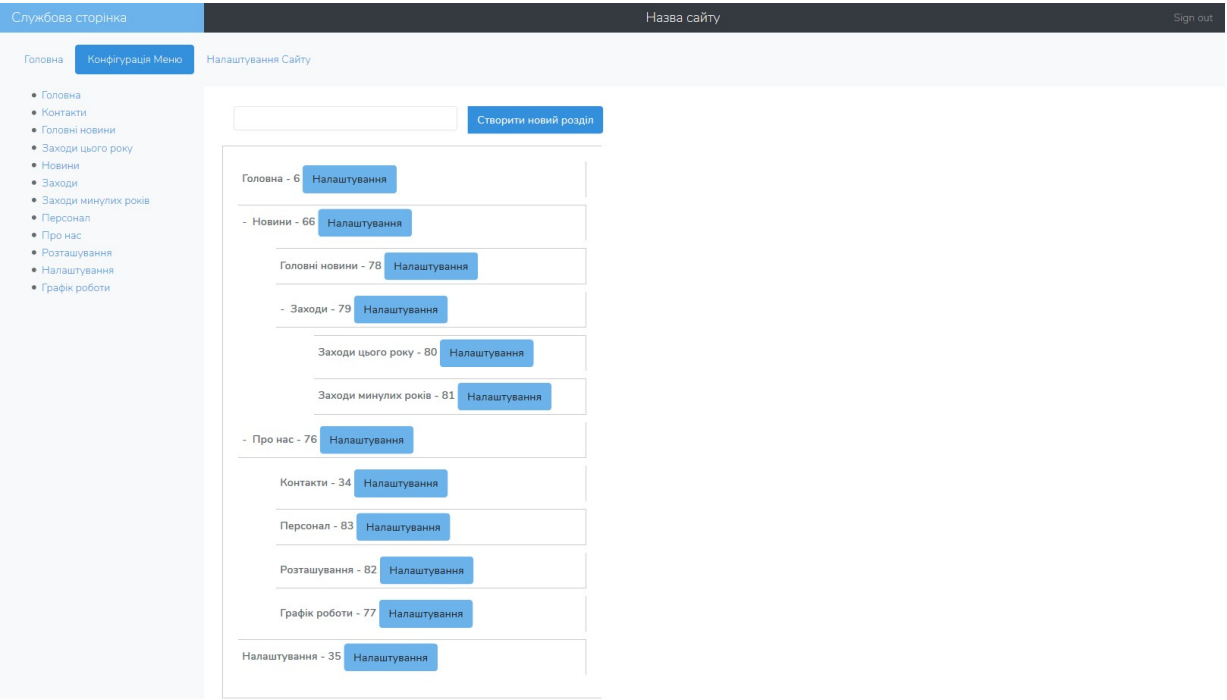


Рисунок 4.4.3 – Сторінка налаштування меню

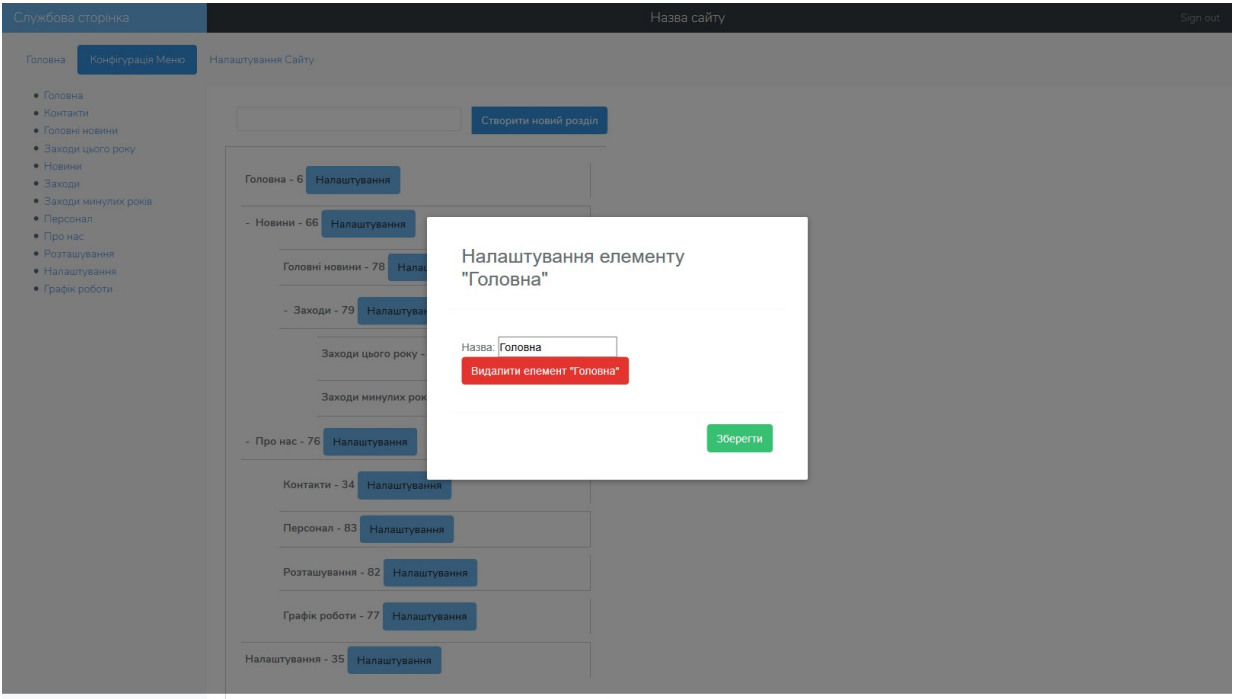


Рисунок 4.4.4 – Вікно конфігурації елементу меню

4.4.4 Розділ налаштування сайту службової сторінки

Даний розділ (рис 4.4.5) призначений для налаштування головних параметрів сайту. Розділ розбитий на декілька окремих вікон. Розглянемо кожне з них:

- Статус сайту. Містить інформацію про стан сайту (увімкнений чи вимкнений) та кнопку, за допомогою якої відбувається зміна цього стану.
- Головні налаштування. Містить декілька конфігураційних параметрів, а саме: Заголовок – задає основну назву сайту, Короткий заголовок – задає, що буде відображатися у закладці веб-браузера, Ключові слова – слова, які полегшують пошук сайту у пошукових системах.
- Додаткові налаштування. Містить два параметри: Кількість постів у списку – задається кількість постів, що будуть одночасно відображені на сторінці, Формат часу – встановлює дванадцяти або двадцяти чотирьох годинний формат часу на сайті.
- Налаштування формату зображень. Містить налаштування розміру малих, середніх та великих зображень у пікселях.
- Кнопка «Зберегти Налаштування» призначена для збереження встановлених параметрів до бази даних.

Службова сторінка

Назва сайту

Sign out

Головна

Конфігурація Меню

Налаштування Сайту

- Головна
- Контакти
- Головні новини
- Заходи цього року
- Новини
- Заходи
- Заходи минулих років
- Персонал
- Про нас
- Розташування
- Налаштування
- Графік роботи

Статус сайту: Вимкнений

Увімкнути

Головні налаштування:

Заголовок:

Кафедра

Короткий заголовок:

RTPS

Ключові слова:

RTPS RTF KPI

Додаткові налаштування:

Кількість постів у списку:

5

Формат часу:

☐ 12 годин

☒ 24 години

Налаштування формату зображень:

Розмір малих зображень:

100

Приклад Зображення

Розмір середніх зображень:

300

Приклад Зображення

Розмір великих зображень:

700

Приклад Зображення

Зберегти налаштування

Рисунок 4.4.5 – Сторінка налаштування сайту

4.4.5 Розділ налаштування вмісту пункту меню

Даний розділ призначений для створення та налаштування вмісту конкретного пункту меню. Тобто тут створюється контент, який буде бачити фінальний користувач при переході на даний пункт меню.

Все починається з вибору пункту меню, що буде налаштовуватись. Далі відкривається сторінка налаштування на якій в першу чергу обирається необхідний тип розділу (рис. 4.4.6), потім обирається необхідний тип сторінки, а саме проста, проста плюс, галерея, текст плюс фото та контакти (рис. 4.4.7). Відповідно при виборі необхідного типу сторінки відкривається вікно необхідних налаштувань, рисунки даних вікон наведені на рис. 4.4.8 та рис.4.4.9.

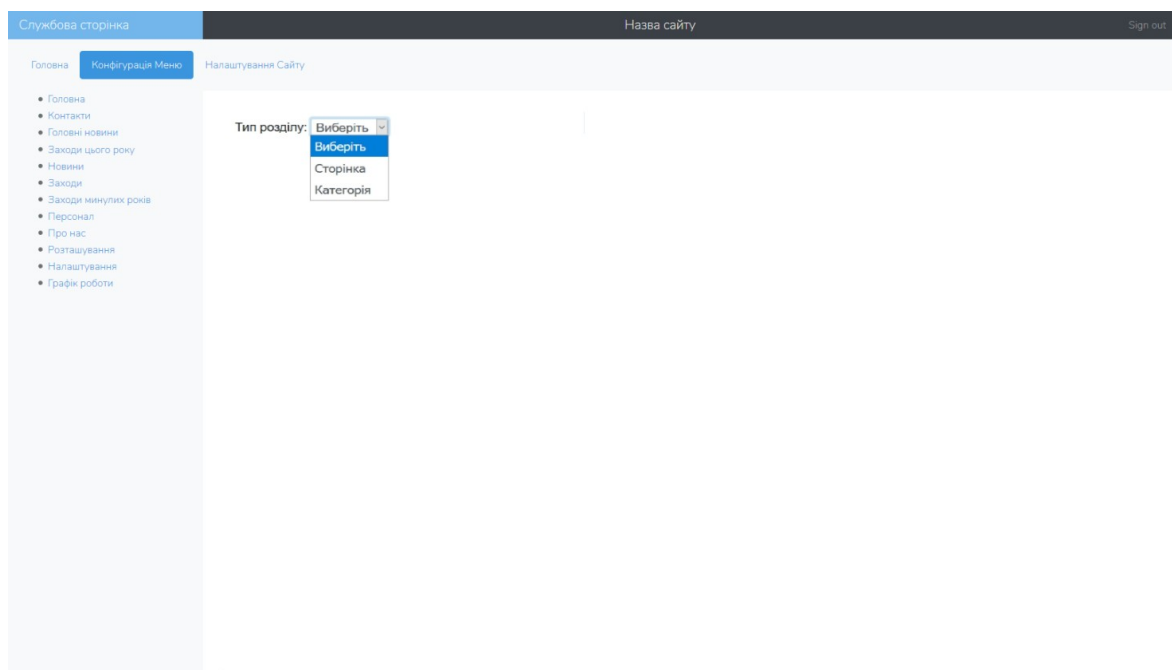


Рисунок 4.4.6 – Вибір типу розділу

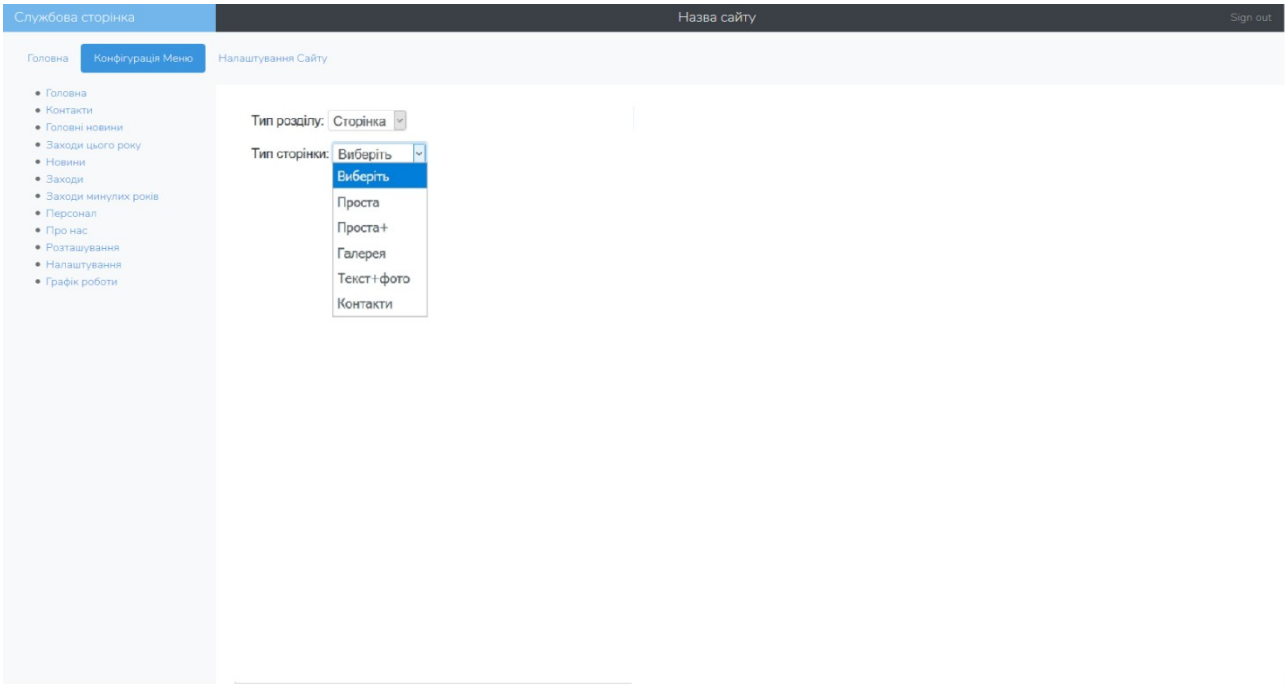


Рисунок 4.4.7 – Вибір типу сторінки

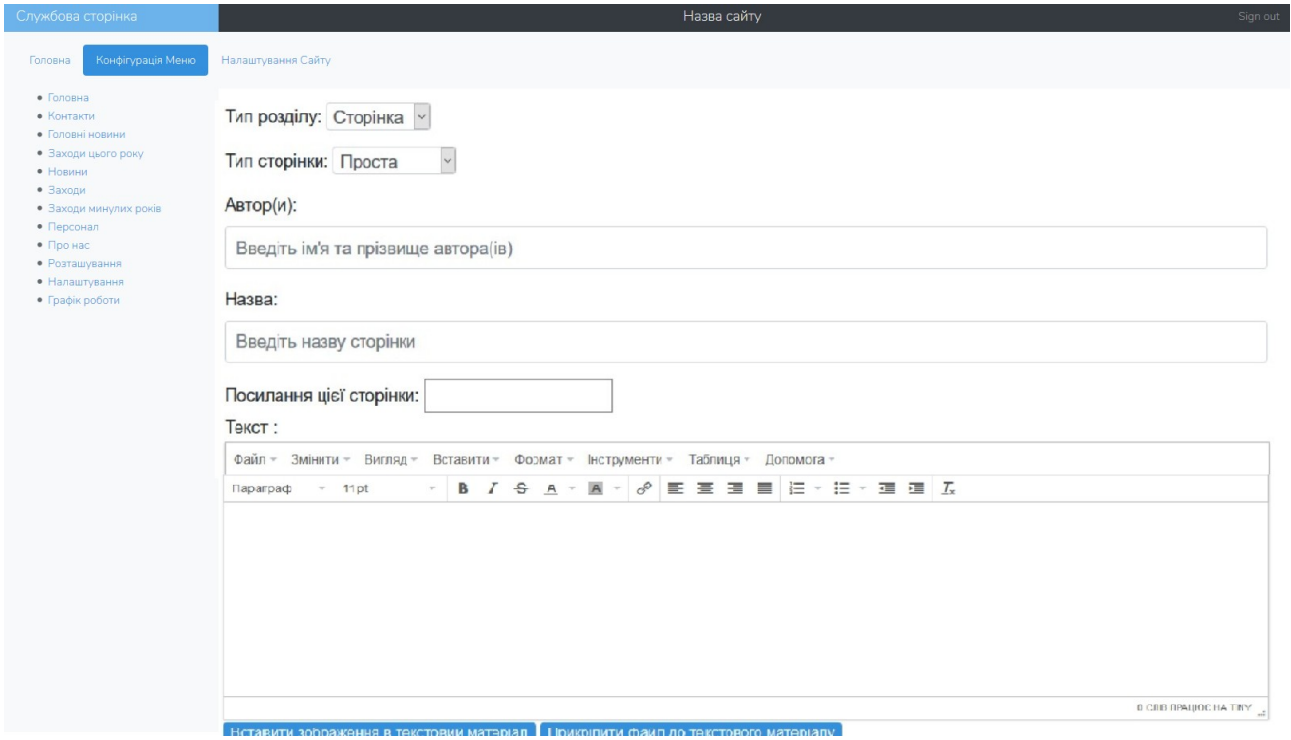


Рисунок 4.4.8 – Тип сторінки проста

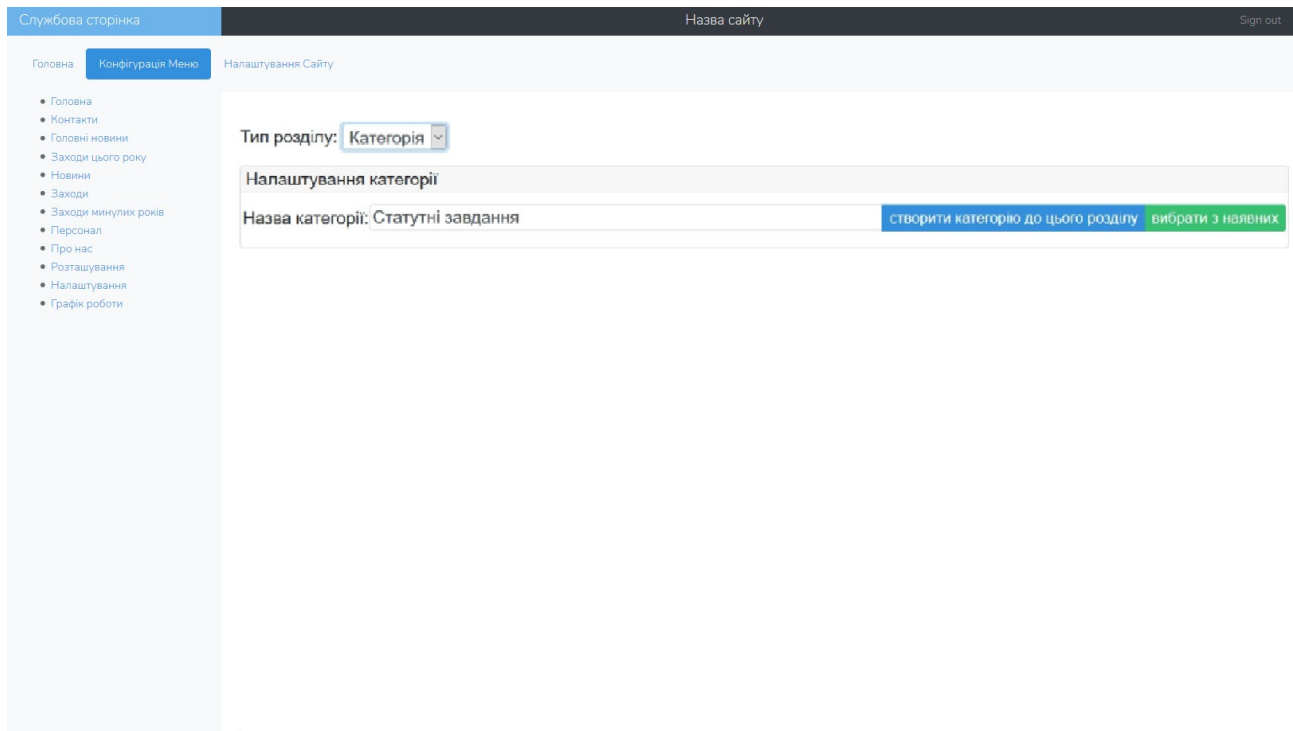


Рисунок 4.4.9 – Тип розділу категорія

4.5 Файлова структура програмного забезпечення

Оскільки програмне забезпечення виконано на основі фреймворку Laravel, то відповідно файлова структура формується на його основі, а саме складається з таких ключових компонентів:

- App.css після компіляції проекту містить в собі всі файли css
- App.js після компіляції проекту містить в собі всі файли js
- Blade.php є головною сторінкою продукту
- Web.php містить в собі логіку маршрутизації

Більш детально ці та інші файли описані в табл.4.5.1.

Таблиця 4.5.1 – Опис файлової структури програми

№ п/п	Назва файлу	Призначення файлу
1	cms.blade.php	Головна сторінка
2	web.php	Описує маршрутизацію всього проекту

3	app.js	Описує внутрішню маршрутизацію Vue.js
4	backgroundControlPanel.vue	Компонент заднього фону сторінки налаштувань
5	menu.vue	Компонент налаштування меню
6	navbarLeft.vue	Компонент лівої панелі
7	navbarTop.vue	Компонент верхньої панелі
8	MenuController.php	Описує функції компонента меню з базою даних
9	PostController.php	Описує функції компонента пости з базою даних
10	SiteController.php	Описує функції компонента сайт з базою даних
11	SiteSetController.php	Описує функції компонента налаштування сайту з базою даних
12	AuthSiteId.php	Перевіряє авторизацію користувача
13	Menu.php	Описує зв'язок компонента меню з базою даних
14	Option.php	Описує зв'язок компонента опції з базою даних
15	Post.php	Описує зв'язок компонента пости з базою даних
16	Site.php	Описує зв'язок компонента сайт з базою даних

4.6 Опис алгоритму створення контенту

Алгоритм роботи системи створення контенту показаний на рис.4.6.

Обравши необхідний пункт меню, користувач потрапить на сторінку налаштування вмісту обраного пункту меню. Спершу необхідно обрати тип сторінки (сторінка чи категорія). Якщо обрана «категорія», то на сторінці з'явиться конфігураційне меню, в якому можна обрати назву категорії та задати які сторінки будуть на ній відображатися.

Якщо обрано було «сторінка», то з'явиться ще одне вікно вибору, де буде необхідно обрати бажаний тип сторінки (проста, проста плюс, галерея, текст плюс фото або контакти). В залежності від вибору на сторінці

завантажитися необхідне вікно створення та налаштування сторінки бажаного типу.

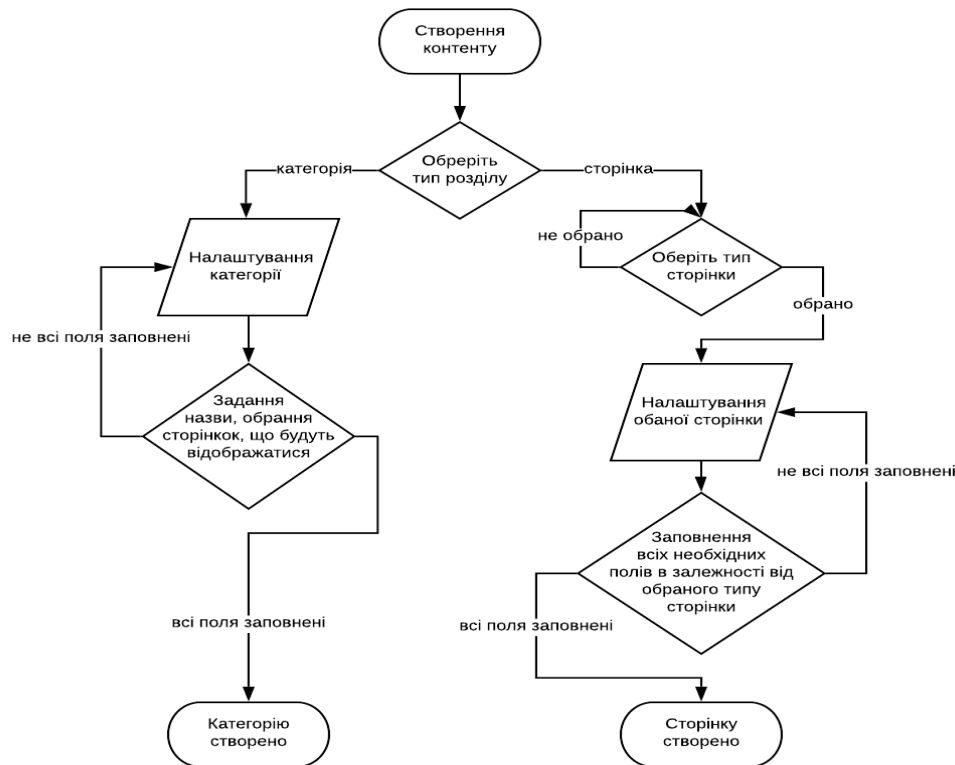


Рисунок 4.6 – Алгоритм створення контенту

Висновки

За результатами цього розділу були проаналізовані та обрані необхідні інструменти для зручнішої та швидшої розробки програмного забезпечення, також був обраний інструментарій для зручного адаптивного та динамічного представлення інформації. Була проаналізована та розроблена інфологічна та датологічна системи продукту. Розроблені необхідні структури бази даних та взаємозв'язки між її компонентами, що дало змогу уникнути дублювання інформації та як результат пришвидшення роботи. Було розроблено конфігурацію, зовнішній вигляд і функціонал службової сторінки з усіма її компонентами (головна, налаштування меню, налаштування сайту) та

конфігурацію, зовнішній вигляд і функціонал сторінки налаштування зовнішнього вигляду.

5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

Так як метою даної дипломної роботи є розробка програмного забезпечення для динамічного адаптивного представлення інформації на основі інформаційно-комунікаційної системи «ПОЛІДАР», то в цьому розділі буде доцільним першочергово розглянути питання, що пов'язані з забезпеченням безпечних та комфортних умов праці при використанні засобів обчислювальної техніки, а також питання з електробезпеки та безпеки в надзвичайних ситуаціях.

У зв'язку з цим у даному розділі з урахуванням вимог ДСанПіН 3.3.2.007-98 та НПАОП 0.00-7.15-18 визначені основні потенційно небезпечні і шкідливі виробничі фактори, що виникають при експлуатації ПЕОМ, а також запропоновані технічні рішення та організаційні заходи з безпеки і гігієни праці і визначені основні заходи з безпеки надзвичайних ситуацій .

5.1 Визначення основних потенційно небезпечних та шкідливих факторів

За даними Всесвітньої організації охорони здоров'я професійна діяльність користувача ПК може в окремих випадках приводити до порушення функцій зорових аналізаторів, кістково - м'язової системи (примусова поза) і порушень, зв'язаних зі стресовими ситуаціями і нервово - емоційною напругою при роботі.

Комп'ютерна техніка, що використовується, є сучасною технікою, виконаною з дотриманням багатьох сучасних норм та вимог охорони праці. Зокрема, монітори мають тип LR/NI. Тип (Low Radiation) має низький рівень

випромінювання екрана монітора. Тип NI (Non - Interlaced) має порядкове розгорнення, що сприяє меншому стомленню очей при роботі з відео монітором.

Відповідно до ДСанПіН 3.3.2.007 - 98 основними шкідливими та небезпечними виробничими факторами, які зв'язаними з роботою на ПЕОМ є:

- електромагнітне випромінювання ВДТ, ПЕОМ;
- можливість враження електричним струмом;
- шуми, які пов'язані з роботою охолоджувальної системи комп'ютера;
- велика напруга зорових органів, як наслідок втома користувача ПЕОМ;
- навантаження на кисті рук і пальці, що за відсутності профілактичних заходів та регулярного контролю, здатне спричинити професійні захворювання;
- тривале перебування в положенні сидячи без значної зміни самого положення, що здатне викликати застійні явища в організмі людини;
- електромагнітне випромінювання діапазону;
- наявність іонізуючого рентгенівське випромінювання (НРВ);
- випромінювання оптичного діапазону (ультрафіолетове, інфрачервоне та випромінювання видимого діапазону);
- електростатичне поле;
- відблиски на екрані монітора.

5.2 Оцінка існуючих умов

При роботі з ПЕОМ виконуються усі необхідні вимоги, а саме:

- приміщення не розміщене в підвальному чи цокольному поверсі

- приміщення має природне та штучне освітлення
- приміщення не межує з приміщеннями, в яких рівні шуму і вібрації перевищують допустимі значення (виробничі цехи, майстерні тощо)

Допустимі рівні звуку, еквівалентні рівні звуку і рівні звукового тиску в октавних смугах частот наведені в таблиці 5.1

Таблиця 5.1 – Еквівалентні рівні звуку і рівні звукового тиску.

Вид трудової діяльності, робочі місця	Рівні звукового тиску в дБ									
	в активних смугах із середньо геометричними частотами, Гц									
	31.5	63	125	250	500	1000	2000	4000	8000	Рівні звуку, еквівалентні рівні звуку, дБА\дБАккв
Програмісти ЕОМ	86	71	61	54	49	45	42	40	38	50
Оператори в залах обробки інформації на ЕОМ та комп'ютерного набору	96	83	74	68	63	60	57	55	54	65
В приміщеннях для розташування шумних агрегатів ЕОМ	103	91	83	77	73	70	68	66	64	75

- приміщення має обладнання опалення, кондиціонування повітря
- віконні прорізи приміщення для роботи з ВДТ обладнані регульованими пристроями (жалюзі, завіски, зовнішні козирки).

Світильники загального освітлення

При відсутності світильників серії ЛПО36 з ВЧ ПРА і без ВЧ ПРА модифікації "кососвітло" допускається застосування світильників загального освітлення серії:

ЛПО13 - 2х40/Б - 01;

ЛПО13 - 4х40/Б - 01;

ЛПО13 - 2х40 - 06;

ЛПО13 - 2х65 - 06;

ЛСО05 - 2х40 - 001;

ЛСО05 - 2х40 - 003;

ЛСО04 - 2х36 - 008;

ЛПО34 - 4х36 - 002;

ЛПО34 - 4х58 - 002;

ЛПО31 - 2х31 - 002,

а також їх вітчизняні та зарубіжні аналоги.

Таблиця 5.1 – Норми мікроклімату для приміщень з ВТД ЕОМ та ПЕМ

Пора року	Категорія робіт	Температура повітря, град.С не більше	Відносна вологість повітря, %	Швидкість руху повітря м\с
Холодний	легка-1а	22-24	40-60	0.1
	легка-1б	21-23	40-60	0.1
Теплий	легка-1а	23-25	40-60	0.1
	легка-1б	22-24	40-60	0.2

- поверхня підлоги є рівною, неслизькою, з антистатичними властивостями.
- в приміщенні щоденно проводиться вологе прибирання
- Робочі місця працівників з екранними пристроями мають такі розміри, що надає простір для зміни робочого положення та рухів.

- Усе випромінювання від екранних пристроїв зведене до гранично допустимого рівня (вплив на людину факторів довкілля - шуму, вібрації, забруднювачів, температури тощо, який не спричиняє соматичних або психічних розладів, а також змін стану здоров'я, працездатності, поведінки, що виходять за межі пристосувальних реакцій) з погляду безпеки та охорони здоров'я працівників.
- Організація робочого місця працівника з екранними пристроями забезпечує відповідність усіх елементів робочого місця та їх розташування ергономічним, антропологічним, психофізіологічним вимогам, а також характеру виконуваних робіт.
- Робочий стіл або робоча поверхня мають достатній розмір та поверхню з низькою відбивною здатністю, що допускає гнучкість під час розміщення екрана, клавіатури, документів і відповідного устаткування.
- Робоче крісло є стійким і дозволяє працівнику з екранними пристроями легко рухатися та займати зручне положення.

5.3 Комплекс вправ для відновлення

Оскільки робота з програмним забезпеченням передбачає знаходження перед персональним комп'ютером протягом тривалого часу, тому рекомендовано проводити такі фізичні вправи:

Комплекс вправ для очей

Вправи виконують, відвернувшись від екрана.

Варіант 1

1. Вихідне положення (В.п.) - сидячи, руки на колінах. Закрити очі, сильно напруживши очні м'язи, на рахунок "раз-шість", потім відкрити очі,

подивитись вгору на рахунок "сім-вісім", подивитись на рахунок "дев'ять-десять". Повторити 5 разів.

2. В.п. те саме. Робити колові рухи очима, фіксуючи погляд в таких положеннях: додолу-вліво-вгору-вправо-додолу. Повторити 5 разів. Потім те саме 5 разів у зворотному напрямі.

3. В.п. те саме. Закрити очі на рахунок "раз-два", відкрити очі і подивитися на кінчик носа на рахунок "три-чотири". Повторити 5 разів.

Варіант 2

1. В.п. - сидячи. Швидко моргати очима протягом 15 с.

2. В.п. - сидячи завдальшки 30-35 см од вікна обличчям до нього. Дивитися на позначку на шибці протягом 5 с, потім перевести погляд на більш віддалений об'єкт за вікном і дивитися ще протягом 5 с. Повторити 10 разів.

3. В.п. - сидячи. Швидко перевести погляд по діагоналі: праворуч вгору - ліворуч униз. Потім дивитися прямо у далеч протягом 6 с. Швидко перевести погляд по діагоналі: ліворуч вгору - праворуч униз. Потім дивитися прямо у далеч протягом 6 с. Повторити 4-5 разів.

Комплекс вправ для поліпшення мозкового кровообігу

Нахили і повороти голови справляють механічну дію на стінки шийних кровоносних судин, підвищуючи їх еластичність. Тренування вестибулярного апарату сприяє розширенню кровоносних судин головного мозку, а дихальні вправи, особливо дихання через ніс, збільшують їх кровонаповнення. Все це підсилює мозковий кровообіг, тим самим полегшуючи розумову діяльність.

Варіант 1

1. В.п. - основна стійка (о.с.). На рахунок "раз" - руки за голову, лікті розвести, голову нахилити назад. На рахунок "два" - лікті вперед. На рахунок "три-чотири" - руки розслаблено опустити вниз, голову нахилити вперед. Повторити 4-6 разів у повільному темпі.

2. В.п. - стійка "ноги порізня", пальці стиснуті в кулаки. На рахунок "раз" - різкий мах лівою рукою назад, правою - вгору назад. На рахунок "два" - різко змінити положення рук. Повторити 6-8 разів у середньому темпі.

3. В.п. - сидячи на стільці. На рахунок "раз-два" - плавно відвести голову назад, на рахунок "три-чотири" - голову нахилити вперед, плечі не піднімати. Повторити 4-6 разів у повільному темпі.

Варіант 2

1. В.п. - стоячи або сидячи, руки на поясі. На рахунок "раз-два" коловим рухом відвести праву руку назад з поворотом тулуба і голови праворуч, на рахунок "три-чотири" - те саме ліворуч. Повторити 4-6 разів у повільному темпі.

2. В.п. - стоячи або сидячи, руки в сторони, долоні вперед, пальці розведені. На рахунок "раз" обхопити себе за плечі руками якомога міцніше і далі, на рахунок "два" повернутися у в.п. Повторити 4-6 разів у швидкому темпі.

3. В.п. - сидячи на стільці, руки на поясі. На рахунок "раз" повернути голову праворуч, на рахунок "два" - в.п. Те саме - ліворуч. Повторити 6-8 разів у повільному темпі.

Комплекс вправ для рук

Вправи можна робити в будь-який час протягом дня, спочатку по 2-3 рази, поступово збільшуючи навантаження до 6-10 разів.

Варіант 1

1. Руки, не напружуючи, простягнути вперед на ширину плечей. Повільно згинати й розгинати пальці. Потім з того самого положення повільно згинати і розгинати руки в зап'ястках.

2. Руки простягнути вперед на ширину плечей долонями догори. Згинати і розгинати руки в ліктьових суглобах.

3. Руки опущені вздовж тулуба долонями всередину, пальці без напруження стиснути в кулак. Обертати кулаки за годинниковою стрілкою і проти. З того самого положення згинати і розгинати руки в зап'ястках.

4. Підняти руки в сторони до рівня плечей, потім опустити. Підняти руки в сторони до рівня плечей і обертати їх у плечових суглобах спочатку назад, потім - вперед.

5. Підняти руки вгору і плавно опустити їх вниз.

Варіант 2

1. Масажувати пальцями кісті з тильного і зовнішнього боків.

2. Пальцями правої руки обхопити пальці лівої руки і обережно відхилити їх до зап'ястка. Затримати у такому положенні протягом 5 с. Те саме зробити з пальцями правої руки.

3. Пальцями правої руки обережно гойдати великий палець лівої руки, і навпаки.

4. Міцно стиснути пальці в кулак, потім розтиснути їх і розчепірити.

Комплекс вправ для хребта

Спеціальні вправи для тренування і розслаблення хребта поліпшують периферійний кровообіг, сприяють збереженню правильної постави, оптимальному перерозподілу м'язового навантаження при роботі, цілюще впливають на функціональний стан внутрішніх органів, нервової системи, органів зору, допомагають подолати наслідки гіподинамії.

Виконуючи вправи, не слід робити різких рухів. Треба весь час зважати на свої фізичні можливості. Амплітуду збільшувати поступово. Спочатку кожен вправу слід виконувати 2-5 разів, поступово збільшуючи навантаження до 10 разів. Комплекс вправ слід робити щодня до помітного поліпшення самопочуття. Надалі - досить

двох разів на тиждень.

1. В.п. - лежачи на животі обличчям додолу, ноги на ширині плечей, руки простягнути вперед. Підняти таз і вигнути спину. Коліна і лікті випрямлені. Упор тільки на долоні і пальці ніг. Голова опущена. Опустити таз. Підняти голову і відхилити її назад.

2. В.п. - те саме. Підняти таз і вигнути спину. Руки і ноги прямі. Повільно повернути таз якомога далі вліво, опускаючи лівий бік якомога нижче. Те саме зробити в інший бік.

3. В.п. - сидючи на підлозі, обпираючись на розставлені позаду руки, ноги зігнуті в колінах. Швидко підняти таз і все тіло до горизонтального рівня. Повернутися у в.п.

4. В.п. - лежачи на спині, ноги витягнені. Зігнути ноги в колінах, підтягнути їх до грудей, обхопити руками. Не віднімаючи рук, відхилити ноги від грудей, водночас намагаючись торкнутися підборіддям колін. Залишатися в такому положенні протягом 5 с.

5. В.п. - лежачи на животі обличчям вниз. Підняти таз і вигнути спину, опустивши голову і обпираючись на випрямлені руки і ноги. В такому положенні обійти кімнату.

5.4 Електробезпека

Все наявне в лабораторії електроустаткування можна віднести до I (системні блоки) та II (ВДТ) класів щодо електрозахистом.

Помешкання лабораторії по рівню небезпеки поразки людей електричним струмом можна віднести, до помешкань без підвищеної небезпеки, тому що:

- відносна вологість повітря не перевищує 75%;

- покриття підлоги (лінолеум) є діелектриком;
- температура повітря не перевищує значень, більших ніж 35 °С;
- відсутня можливість одночасного доторку людини до з'єднаних із землею металоконструкцій будівлі, технологічних апаратів, механізмів і т.п., з одного боку, і до металевих корпусів електроустаткування - з іншого боку;
- відсутні хімічно агресивні середовища.

Електромережа трьохфазна з глухо заземленою нейтраллю, із зануленням і повторним заземленням нульового проводу відповідно до вимог ПБЕ та ПУЕ. Мережа обладнана автоматом струмового захисту, розрахованого на струм 10А. Час спрацювання автомату 0,1с.

Для того, щоб не допустити ураження людини електричним струмом при виникненні аварійних ситуацій необхідно заземлити все обладнання, що працює від мережі 220В, 50Гц. Опір нульового проводу повинен бути таким, щоб при замиканні на корпус або нульовий провід виникав струм короткого замикання, сила якого повинна перевищувати в 1,4 рази номінальний струм спрацювання автомата струмового захисту (при струмі короткого замикання менше 100А).

$$I_{кз} = \frac{U_{\phi}}{R_0 + R_{\phi} + Z_{T/3}},$$

де U_{ϕ} – напруга фази мережі (220 В),

R_0 – опір нульового дроту на ділянці від фазного трансформатора до розетки “Вхід мережі” (~3 Ом),

R_{ϕ} – опір фазного дроту на тій же ділянці (~3 Ом).

$Z_{T/3}$ – еквівалентний опір транзистора (0.12 Ом)

Підставимо ці значення в формулу та підрахуємо:

$$I_{кз} = \frac{220}{3+3+0.12} = 35,95 \text{ A}$$

$$K = \frac{I_{кз}}{I_{авт}} = \frac{35,95}{10+0.12} = 3,55$$

Таким чином струм короткого замикання при виникненні аварійної ситуації в 3,55 рази перевищує номінальний струм спрацювання автомата, що задовольняє встановленим нормам.

Опір заземлюючих пристроїв не перевищує значень встановлених ГОСТ 12.1.030-81.

5.5 Безпека в надзвичайних ситуаціях

Безпека в НС регламентується ПЛАС(НПАОП 0.00-4.33-99). Основними складовими частинами ПЛАС є розробка технічних рішень та організація заходів щодо оповіщення, евакуації та дій персоналу при виникненні НС, а також вирішення питань з пожежної безпеки.

5.6 Пожежна безпека

Відповідно до ДСТУ Б В.1.1-36:2016 робоче приміщення лабораторії відноситься до категорії В по вибухопожежній небезпеці. Відповідно до НПАОП 40.1-1.32-01 клас робочих зон приміщення лабораторії по пожежонебезпеці - П-Па. Можливими причинами пожежі в приміщенні є несправність електроустаткування, коротке замикання проводки, і порушення протипожежного режиму (використання побутових нагрівальних приладів, паління).

У зв'язку з цим, відповідно до вимог ПБЕ та ПУЕ, необхідно передбачити наступні заходи.

- постійний контроль стану засобів пожежогасіння;
- застосування автоматичних установок пожежної сигналізації;
- організація за допомогою технічних засобів, включаючи автоматичні, своєчасного оповіщення та евакуації людей.
- контроль за станом ізоляції струмоведучих дротів;
- заборонено паління в приміщенні;
- неприпустимість знаходження у приміщенні горючих і вибухонебезпечних речовин;
- допуск до роботи осіб, які в установленому порядку пройшли навчання, інструктаж і перевірку знань з пожежної безпеки.

Для гасіння пожежі в робочому приміщенні лабораторії (клас пожежі „Е”- наявність електрообладнання під напругою) використовуються вогнегасники ВП-1 — “Момент” (2 шт.). Додатково в коридорі розташовані вогнегасники ВХП-10. Також на сходовій клітці розташований пожежний кран. Така кількість первинних засобів пожежогасіння відповідає вимогам ДСТУ3675-98 та ISO3941-77, якими передбачене обов'язкова наявність двох вогнегасників до 100 м² площі підлоги для приміщення типу конструкторське бюро.

Згідно вимог ДБН В.2.5-56:2014 робоче приміщення лабораторії необхідно оснастити системою автоматичної пожежної сигналізації.

Будинок має два евакуаційних виходи: через головний хід і додатковий евакуаційний вихід. Шляхи евакуації відповідають установленим нормам. Двері відкриваються назовні. Коридор веде до двох сходових кліток, одна з яких виходить безпосередньо на вулицю, а друга має вихід на вулицю через вестибюль і головний вхід. Сходові клітки виконані з не пальних матеріалів. Сходи мають природне бічне освітлення і штучне евакуаційне освітлення.

Сходові площадки ширше коридорів. Усі співробітники ознайомлені з планом евакуації.

Дотримано усі вимоги СНиП 2.09.02-85 по вогнестійкості будинку і ширині евакуаційних проходів і виходів із приміщень назовні. Значення основних параметрів шляхів евакуації приведені в табл. 8.4.3.

Таблиця 5.3. - Характеристики і норми еваковиходів

Параметр	Фактичне значення	Норма
Висота дверних прорізів	2,0 м	Не менше 2 м
Ширина дверних прорізів	0,8 м	Не менше 0,8 м
Ширина проходу для евакуації	Більше 1,5 м	Не менше 1 м
Ширина коридору	2 м	Не менше 2 м
Число виходів з коридору	2	Не менше 2
Ширина сходового маршу	1,2 м	Не менше 1 м
Висота поруччя сходів	1 м	Не менше 0,9 м

У приміщенні є план евакуації. Мінімальний час евакуації в разі виникнення пожежі відповідає вимогам СНиП 2.01.02-85, а максимальне видалення робочих місць від евакуаційних виходів вимогам СНиП 2.09.02-85.

У приміщенні виконуються усі вимоги по пожежній безпеці відповідно до вимог НАПБ А.01.001-2014 «Правила пожежної безпеки в Україні». Нижче приводиться план евакуації у разі виникнення НС (пожежі)

Висновки

У процесі виконання дипломної роботи вдалося здійснити необхідний обсяг робіт, зокрема було зроблено наступне:

1. Вивчена проблема представлення інформації;
2. Поставлено завдання для її вирішення;
3. Проведено аналіз існуючого програмного забезпечення, визначено їх переваги і недоліки, які виправити можливо лише через переписування значних фрагментів програмного коду самих систем.;
4. Визначено вимоги до створення програмного забезпечення, яке повинно мати логічну структуру, інтуїтивно зрозумілий інтерфейс;
5. Розроблено структуру програмного забезпечення, інфологічну та датологічну модель;
6. Розроблено програмне забезпечення, яке містити в собі всі необхідні інструменти для динамічного адаптивного представлення контенту.

Як можливий варіант розвитку програмного забезпечення – розширення його функціоналу. Основним є розробка додаткових шаблонів відображення контенту. Також бажаним є створення більшої кількості заготовок зовнішнього вигляду сторінки програмного забезпечення.

Також є можливість реалізувати підрахунок і обробку статистики відвідувань сайту, якісний і кількісний аналіз цієї інформації. Це допоможе користувачам оцінити, наскільки ефективним є представлення їхньої інформації.

Дане програмне забезпечення може бути використане для зручного зберігання та відображення лекційних матеріалів, презентацій, фото та відеолекцій. Це все може бути використане для дистанційного навчання студентів. Також програмне забезпечення дає змогу реалізувати систему зворотного зв'язку (коментарі), що в свою чергу дасть змогу поліпшити контент, що відображається на даній сторінці.

ПЕРЕЛІК ПОСИЛАНЬ

1. Worldwide digital population: <https://www.statista.com/statistics/617136/digital-population-worldwide/>
2. Usage statistics of content management systems: https://w3techs.com/technologies/overview/content_management
3. Система Wix: <https://www.wix.com/account/sites>
4. Система Бітрікс24: <https://www.bitrix24.ua/>
5. Система uKit: <https://ukit.com/ru>
6. Система Google Сайти: <https://sites.google.com/new>
7. Система Vigbo: <https://vigbo.com/>
8. Система Tilda: <https://tilda.cc/ru/>
9. Товкач, І.О. Інформаційно-комунікаційна система «ПОЛІДАР» [Текст] / І.О. Товкач // Інформатика, математика, автоматика: матеріали та програма науково-технічної конференції, м. Суми, 18-22 квітня 2016 р. / Відп. за вип. С.І. Проценко. - Суми: СумДУ, 2016. - С. 76.
10. Model View Controller: <https://hackernoon.com/beginners-guide-to-ruby-on-rails-mvc-model-view-controller-pattern-4z19196a>
11. The Vue Instance: <https://vuejs.org/v2/guide/instance.html>
12. Reactivity in Depth: <https://vuejs.org/v2/guide/reactivity.html>
13. Vuetify Viewport Breakpoints: <https://vuetifyjs.com/en/components/grids/>
14. dbdiagram: <https://dbdiagram.io>